

Евроазиатская научно-промышленная палата

**МАТЕРИАЛЫ МЕЖДУНАРОДНОЙ
СТУДЕНЧЕСКОЙ НАУЧНОЙ КОНФЕРЕНЦИИ
«СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ»**

ТОМ XIX

Москва
2026

Материалы Международной студенческой научной конференции «Студенческий научный форум 2026» [под ред. к.и.н. Старчиковой Н.Е., отв. секретарь Нефедова Н.И.]. – М.: Издательство Евроазиатской научно-промышленной палаты, 2026. – Том XIX. – 136 с.

ISBN 978-5-6048985-6-7

Международный студенческий научный форум привлекает все больше участников из разных уголков России, а также стран ближнего и дальнего зарубежья. Каждый следующий студенческий форум расширяет не только географические, но и тематические границы, затрагивая самые актуальные проблемы всех отраслей современной науки. Сегодня можно говорить о том, что Международный студенческий научный форум стал одной из наиболее масштабных и представительных студенческих научных конференций, проводимых на территории стран СНГ. Лучшие студенты и руководители секций (ученые и преподаватели ВУЗов) выступают с докладами на конференции, которую проводит Международная ассоциация ученых, преподавателей и специалистов в Московском государственном университете имени М.В. Ломоносова.

В данном сборнике представлены материалы Международной студенческой конференции «Студенческий научный форум 2026».

ISBN 978-5-6048985-6-7

© Евроазиатская научно-промышленная палата

СОДЕРЖАНИЕ

XVIII МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ «СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ 2026»

Медицинские науки

БРУКСИЗМ В СОВРЕМЕННОЙ СТОМАТОЛОГИИ

Иванюк Е. С., Луцук М. В., Остроухова О. Н.

6

Политические науки

ГЕОПОЛИТИЧЕСКИЕ ПОЗИЦИИ РОССИИ В МИРОВОЙ ПОЛИТИКЕ XXI ВЕКА

Горшкова Е. А., Матвеев О. В.

9

Технические науки

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ АВТОМАТИЗИРОВАННОЙ ПРОВЕРКИ НОРМАЛЬНОСТИ РАСПРЕДЕЛЕНИЯ И СТАТИСТИЧЕСКОГО АНАЛИЗА БИОМЕТРИЧЕСКИХ ДАННЫХ

Ананченко И. В., Жижина К. П., Мельников И. Д.

13

ВЛИЯНИЕ МУЛЬТИМЕДИА НА СОВРЕМЕННОЕ ОБРАЗОВАНИЕ

Антоненко А. С., Окулова Е. А.

15

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ТУРИСТИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ

Бабенко Н. Н.

17

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ РЕЧИ: СОСТОЯНИЕ И НАПРАВЛЕНИЯ РАЗВИТИЯ

Бахарев Б.А.

19

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НАТИВНОЙ И КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Гринин А. О.

21

АСПЕКТЫ ИСПОЛЬЗОВАНИЯ МУЛЬТИМЕДИЙНЫХ ТЕХНОЛОГИЙ В ИСКУССТВЕ

Гриценко М. С., Окулова Е. А.

23

РОЛЬ И МЕСТО ТКINTER В СОВРЕМЕННЫХ РЕАЛИЯХ СОЗДАНИЯ КЛИЕНТСКОЙ ЧАСТИ ТЕХНОЛОГИЧЕСКИХ ПРОДУКТОВ

Гурина В. М.

27

ГИБРИДНАЯ АРХИТЕКТУРА AI-АССИСТЕНТА ДЛЯ УПРАВЛЕНИЯ ФИНАНСАМИ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ И ПОВЕДЕНЧЕСКОЙ ЭКОНОМИКИ

Денисов Е. Д.

30

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ОБЪЁМА ВЫПУСКА ПРОДУКЦИИ ПРЕДПРИЯТИЯ

Ермоленко С. В., Мозговенко А.А.

32

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИНХРОННЫХ И АСИНХРОННЫХ ВЕБ-СЕРВЕРОВ

Журавлёв Д. В., Букреев Д. А.

36

АНАЛИЗ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ РАЗРАБОТКИ И ПРОЕКТИРОВАНИЯ ВЕБ-РЕСУРСА <i>Задорожный А. Я., Букреев Д. А.</i>	39
ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ МОДУЛЬНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ <i>Иваненко О. А., Букреев Д. А.</i>	44
ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ В ПОСТРОЕНИИ Е-ПРОФИЛЯ СТУДЕНТА <i>Иванив О. С., Строкань О. В.</i>	48
МЕТОДЫ И МОДЕЛИ ПРОГНОЗИРОВАНИЯ СПОРТИВНЫХ СОБЫТИЙ <i>Константинов А. В., Букреев Д. А.</i>	52
СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ YOLOV8-YOLO11 В ЗАДАЧАХ ОБНАРУЖЕНИЯ ОБЪЕКТОВ В РЕАЛЬНОМ ВРЕМЕНИ <i>Корж А. А., Олейник Н. П.</i>	56
ОБЗОР РАЗВИТИЯ И ТЕХНОЛОГИЧЕСКИХ ВЫЗОВОВ ПЛАТФОРМ ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО СПОРТИВНОМУ ПРОГРАММИРОВАНИЮ <i>Кремь А. А., Мозговенко А. А.</i>	58
ПУЗЫРЬКОВАЯ СОРТИРОВКА: СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМА НА СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ В ЧИТАЕМОСТИ КОДА <i>Кузнецова В. А.</i>	61
КОНФИДЕНЦИАЛЬНОСТЬ ПЕРСОНАЛЬНЫХ ДАННЫХ В СОЦИАЛЬНЫХ СЕТЯХ: РИСКИ И СПОСОБЫ ИХ МИНИМИЗАЦИИ <i>Кузьменко Р. Д.</i>	65
КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА ВЕБ-ИНТЕРФЕЙСА УПРАВЛЕНИЯ УМНЫМ ДОМОМ <i>Кузьменко И. А., Мозговенко А. А.</i>	67
АНАЛИЗ СТАНДАРТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И РАЗРАБОТКА ОПЕРАЦИОННОЙ МОДЕЛИ ДЛЯ ПРЕДПРИЯТИЙ КРИТИЧЕСКОЙ ИНФРАСТРУКТУРЫ <i>Мельников А. В., Мозговенко А. А.</i>	70
КИБЕРБЕЗОПАСНОСТЬ. АНАЛИЗ ВИДОВ ЗАЩИТЫ В ИНФОРМАЦИОННОЙ СРЕДЕ <i>Пелипенко А. П.</i>	72
СТРАТЕГИЯ ПОВЫШЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В БАНКОВСКОМ ОТДЕЛЕНИИ <i>Поздняков И. В., Букреев Д. А.</i>	75
МОДЕЛЬ ЯСТРЕБ-ГОЛУБЬ ДЛЯ АНАЛИЗА АГРЕССИВНОГО И ДРУЖЕЛЮБНОГО ВЕБ-ИНТЕРФЕЙСА <i>Попов Т. И., Покуса Т. В.</i>	79
СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ JAVA И C++ ПРИ СОРТИРОВКЕ ЦЕЛЫХ ЧИСЕЛ И ЗАПИСИ/ЧТЕНИИ ФАЙЛОВ <i>Романенко И. В.</i>	84
АРХИТЕКТУРА МАСШТАБИРУЕМОЙ ЛЕНТЫ НОВОСТЕЙ <i>Романенко В. В., Луцкий Е. А.</i>	88

ОБЗОР МЕТОДОВ ОЦЕНКИ ИНВЕСТИЦИЙ В КИБЕРБЕЗОПАСНОСТИ. ПРОЦЕСС УПРАВЛЕНИЯ РИСКАМИ	
<i>Романюк М. А., Мозговенко А. А.</i>	91
АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ВОЗДУШНОГО ПРОСТРАНСТВА	
<i>Рукас М. К., Букреев Д. А.</i>	92
СОВРЕМЕННОЕ СОСТОЯНИЕ РАЗВИТИЯ ТЕХНОЛОГИЙ ПРОГНОЗИРОВАНИЯ ФИНАНСОВЫХ ОПЕРАЦИЙ	
<i>Соболева И. Р., Букреев Д. А.</i>	97
ЭВОЛЮЦИЯ АДАПТИВНОГО ДИЗАЙНА	
<i>Стрельченко О. В., Олейник Н. П.</i>	100
АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ БАЛАНСА ИГРОВЫХ ХАРАКТЕРИСТИК В АВТОМАТИЗИРОВАННЫХ АНАЛОГАХ НАСТОЛЬНЫХ ИГР	
<i>Сушинский И. В., Лебедев В. А.</i>	103
ЦИФРОВЫЕ ТЕХНОЛОГИИ В СФЕРЕ ОБРАЗОВАНИЯ ЗАПОРОЖСКОЙ ОБЛАСТИ	
<i>Тарасенко И. В.</i>	107
ОБЗОР АЛГОРИТМОВ ДЕТЕКЦИИ ЛИЦ И ИХ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ СИСТЕМЫ УЧЁТА РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКА	
<i>Чернышев О. Г.</i>	113
ОСОБЕННОСТИ МУЛЬТИМЕДИЙНЫХ ПАРКОВ КАК ОБЪЕКТА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ	
<i>Чикарь Л. А., Мозговенко А. А.</i>	117
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ОБЪЁМА ВЫПУСКА ПРОДУКЦИИ ПРЕДПРИЯТИЯ	
<i>Штилов Д. В., Мозговенко А. А.</i>	119
ЗАКОНЫ ХУДОЖЕСТВЕННОГО ПОВТОРА: ПОЧЕМУ UI-ДИЗАЙН НАСТОЯЩЕГО НАПОМИНАЕТ ПРОШЛОЕ	
<i>Якомаскина Т. А., Покуса Т. В.</i>	122
ПРОЦЕСС УПРАВЛЕНИЯ РИСКАМИ И ИНВЕСТИЦИИ В КИБЕРБЕЗОПАСНОСТИ	
<i>Ярошенко Е. А., Мозговенко А. А.</i>	125
<hr/>	
Физико-математические науки	
ПОСТКВАНТОВАЯ КРИПТОГРАФИЯ	
<i>Балакина О. А.</i>	129
<hr/>	
Экономические науки	
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МЕТОДЫ ОПТИМИЗАЦИИ В ЦИФРОВОЙ ЭКОНОМИКЕ	
<i>Мельник Е. Г.</i>	134

**XVIII МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ
«СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ 2026»**

Медицинские науки

**БРУКСИЗМ
В СОВРЕМЕННОЙ СТОМАТОЛОГИИ**

Иванюк Е. С., Луцкич М. В., Остроухова О. Н.

*ФГБОУ ВО «Воронежский государственный
медицинский университет им. Н.Н. Бурденко»,
Воронеж, e-mail: katuivanuyk2007@gmail.com*

В настоящее время наблюдается высокая встречаемость патологических процессов в ротовой полости, причиной которых является физическая деформация тканей зуба, раздражение слизистой оболочки, либо патология височно-нижнечелюстного сустава (ВНЧС) в связи с бессознательным скрежетанием и перемалыванием зубами, не связанным с жеванием или речью. В нормальном состоянии максиллярный и мандибулярные зубные ряды находятся в контакте до 30 минут в день. Однако при развитии патологии время может увеличиться до 40 минут за час. При этом происходит неконтрольный спазм жевательных мышц и смыкание зубов, что ведет к их постоянному стиранию режущими краями, как в дневное время, так и ночью. Такого рода патологические явления получили название бруксизм, исследование причин и симптомов которого, несмотря на мультифакториальную его природу, стали достаточно актуальными в современной стоматологии.

Цель работы – исследование генезиса и этиологии понятия «бруксизм» в современной стоматологии, а также причин его появления.

Материалы и методы: анализ современных литературных данных, а также проведение анкетирования по проблеме бруксизма среди студентов медицинского вуза.

Слово бруксизм имеет греческое происхождение от «brucsein», что в переводе означает «скрежет зубами». Генезис этого понятия положен в 1901 г. в трудах М. Karoly, который исследовал травматическую невралгию и благодаря которому в медицинский оборот было введено понятие «эффект Karolyi» – поражение пародонта при спастическом состоянии жевательных мышц [1]. В 1907 г. M. Marie и M. Piefrwich впервые применили в научной литературе понятие «бруксомания», под которым они понимали стереотипные движения нижней челюсти, сопровождающиеся трением или сжатием зубов [1]. Впервые понятие «бруксизм» было введено S. Miller в 1938 г. [3]. В последующем Е. И. Гаврилов, В. Д. Пантелеев, Л. А. Скорикова и другие российские ученые [3, 4, 6, 7] стали широко применять понятие бруксизма в научных

исследованиях и практике медицины наряду с понятием «парафункция жевательных мышц», под которыми описывали бессознательное механическое перенапряжение в зубочелюстной системе, способствующее не только неблагоприятному прогнозу лечения зубов, но и снижению качества жизни пациентов.

В современной научной литературе существует множество понятий бруксизма, разнообразие которых обусловлено тем, что данная патология рассматривается как окклюзионная дисгармония и элемент стресс-менеджмента, нарушение движения во сне, психическое или поведенческое (соматоформное) расстройство, автономная дисфункция, парафункция, а также побочный эффект медикаментозной терапии.

Семантический анализ литературы позволили выделить некоторые из определений бруксизма:

1) дневная и ночная парафункциональная жевательная активность, которая заключается в скрежетании и постукивании зубами, а также в их трении и стискивании [5, 7];

2) частое проявление генерализованного процесса, характеризующегося разнообразным нарушением и вызванного психологическим стрессом [6, 8];

3) особенность поведения, характеризующегося непроизвольной постоянной (стискивание зубов) или ритмической (скрежетание зубами) двигательной активностью жевательных мышц [9];

4) непроизвольные, неосознанные сокращения жевательной мускулатуры при отсутствии необходимости в пережевывании пищи [10].

R. Slavicek, S. Sato определяют это заболевание как своеобразную функцию борьбы со стрессом, что не требует лечения, а только устранения негативных последствий на ткани зубов [11]. В связи с этим в научных исследованиях обоснован отдельный вид бруксизма – брукс-поведение, при котором не происходит нарушений целостности тканей, что отличает его от бруксизма, связанного с парафункцией жевательного аппарата.

Следовательно, с точки зрения стоматологии, бруксизм выступает нежелательным явлением в зубочелюстной системе: при постоянном повышенном давлении на зубы наблюдается стирание эмали и истончение ткани, в связи с этим доступ микроорганизмов через сколы и микротрещины в подлежащие ткани зуба возрастает, и характерно появление кариеса. В результате повышается чувствительность зубов на хими-

ческие, физические и температурные раздражители (гиперестезия). При постоянном скрежете режущие края зубов стираются, становятся сглаженными, коронковая часть зубов деформируется. Вследствие этого может нарушиться прикус, что повлечет за собой дефекты в речи, мимике и жевании. Повышенное трение зубов негативно сказывается на ношение пломб и протезов, что заметно уменьшает срок их службы. Происходит нарушение в периодонтальной связке зуба – выраженное шатание зубов при патологии, десна уменьшается в объеме (рецессия). При этом могут возникать сильные боли за счет постоянного мышечного напряжения, гипертрофии и деформации, когда заметна асимметрия на лице и аномальные черты. В таких ситуациях пациенты жалуются на боли в области ВНЧС, которые отдают в уши и шею, возникают головкружение и головные боли, что объясняет частоту начальных проявлений бруксизма в работе оториноларингологов или неврологов.

В рамках исследования бруксизма было проведено анкетирование студентов-стоматологов 2 курса ФГБОУ ВО «Воронежский медицинский университет имени Н.Н. Бурденко», которые ответили на вопросы, касающиеся их общесоматического статуса. Среди 50 респондентов на вопрос «Болели ли Вы бруксизмом» положительно ответили 11 человек (22%). При этом 81,8% (9 человек) из переболевших наблюдали соответствующие симптомы в детском возрасте (3-6 лет), остальные (8,2%) – в подростковом (после 12 лет). В настоящий момент опрашиваемые не болеют.

Отвечая на вопрос «Какие симптомы болезни наблюдались во время сна: скрип или стискивание зубами?» большинство (72,7%, или 8 человек) студентов подтвердили скрежет зубами, наряду с которым наблюдались также и другие парафункциональные привычки, такие как покусывание щек или губ. Остальные 3 человека отметили стискивание зубов, сопровождающееся небольшим увеличением слюноотделения.

Спрашивая о том, как можно описать Ваше психологическое состояние во время болезни, респондентам были предложены ответы: стабильное, тревожное, контролируемое. По результатам ответов около трети студентов (4 человека) отметили свое состояние как контролируемое. Большинство опрашиваемых (45,5%) указали на тревожное свое состояние во время болезни, связанное с определенными событиями в жизни. И только 2 человека (18,2%) охарактеризовали свое состояние как стабильное.

Отвечая на вопрос «Болели ли Ваши родители бруксизмом?» практически все студенты ответили «нет». На вопрос «Имелись ли у Вас проблемы с нарушением осанки?» многие (54,5%) указали на наличие проблем с позвоночником и нарушение осанки, лечением которого они занимались регулярно.

Вопрос «Связываете ли появление бруксизма с медикаментозным лечением?» был затруднительным, поскольку сложно выявить влияние каких-либо медицинских препаратов на состояние зубочелюстной системы без соответствующих лабораторных исследований. Однако некоторые респонденты (36,3% или 4 человека) допустили возможные последствия приема лекарств, которые могли вызвать появление признаков бруксизма.

На вопрос «Связываете ли появление бруксизма с посещением стоматолога?» положительно ответило 27,3% студентов, которые стали наблюдать признаки бруксизма после ортопедических процедур в детском возрасте.

Систематизируя причины возникновения бруксизма можно выделить следующие факторы, способствующие его развитию:

1) неврогенные – источником служат неврологические расстройства пациента, проблемы со сном, некоторые расстройства центральной и периферической нервной системы;

2) психогенные – психоэмоциональное состояние является одним из наиболее значимых факторов в развитии бруксизма, поскольку скрежет зубами часто рассматривается как подсознательный механизм физиологического реагирования на стресс или внутреннее перенапряжение;

3) некачественные зубные реставрации, выполненные в результате неправильного пломбирования, протезирования или применения других ортопедических конструкций, которые могут спровоцировать жевательную активность мышц;

4) остеопатические – нарушение осанки, врожденные нарушения или травма позвоночного столба, смещение шейных позвонков;

5) фармакологический эффект, обусловленный воздействием некоторых лекарственных препаратов или веществ на развитие бруксизма (в частности, антидепрессантов, психостимуляторов, кофеина и алкоголя, никотина и пр.);

6) генетическая предрасположенность: если у одного из родителей был бруксизм, вероятность его развития у ребенка значительно возрастает, что указывает на потенциал наследственного компонента в регуляции мышечной активности или реакций на стресс;

7) возрастные особенности, которые способствуют доминированию детского бруксизма среди всех возрастных групп пациентов – у детей бруксизм связан преимущественно с прорезыванием молочных или постоянных зубов, а также с психоэмоциональными переживаниями (стрессом, адаптацией в детском саду, школе). При этом зачастую детский бруксизм носит временный характер и проходит самостоятельно, тогда как у взрослых – более стойкий, имеет многофакторную природу происхождения и требует комплексного подхода к лечению.

Вывод. В стоматологической практике бруксизм выступает нежелательным явлением.

ем в зубочелюстной системе, при котором наблюдается стирание эмали и истончение ткани. Результаты анкетирования показали, что психоэмоциональное состояние является одним из наиболее значимых факторов в развитии бруксизма. Полиэтиологичность и мультифакториальность патогенетических взаимодействий бруксизма определяют важность исследования причин его развития и комплексного подхода к лечению.

Список литературы

1. Marie M. M., Pietkiewicz M. La bruxomania // *Rev. Stomatologia*. 1907. No. 14. P. 107-116.
2. Miller S. C. Text book of periodontia // Philadelphia. 1938. P. 91-93.
3. Гаврилов Е. И., Пантелеев В. Д. Особенности ортопедической помощи больным с парафункциями жевательных мышц // *Стоматология*. 1990. № 5. С. 80-81.
4. Скорикова Л. А. Характеристика совместной функции жевательных мышц и головного мозга у лиц с парафункциями жевательных мышц // *Новое в стоматологии*. 2000. № 7. С. 86-91.
5. Кичигина К. Е. Бруксизм – произвольный скрежет зубами // *Наука, образование, общество: тенденции и перспективы развития : сборник материалов VII Международной научно-практической конференции (Чебоксары, 24 сентября 2017 г.) / Редколлегия: О. Н. Широков и др.* Чебоксары: ООО «Центр научного сотрудничества «Интерактив плюс», 2017. С. 42-43.
6. Орлова О. Р., Алексеева А. Ю., Мингазова Л. Р., Конавалова З. Н. Бруксизм как неврологическая проблема (обзор литературы) // *Нервно-мышечные болезни*. 2018. № 8(1). С. 20-27.
7. Артёмов В. Г., Усманова Ш. Ш. Бруксизм – с позиции аллопатической и холистической медицины // *Мануальная терапия*. 2015. № 1. С. 69-73.
8. Гридина В. О., Каракулова Ю. В. Бруксизм как причина болевого синдрома в области лица // *Российский журнал боли*. 2019. Т. 17, № S1. С. 8-9.
9. Бредик В. С. Бруксизм в современной стоматологии // *Молодой ученый*. 2024. №41(540). С. 78-80.
10. Хайбуллина Р. Р., Герасимова Л. П., Кабирова М. Ф. и др. Современные технологии при лечении пациентов с хроническим генерализованным пародонтитом и бруксизмом // *Российский журнал биомеханики*. 2016. Т. 20, № 4. С. 316-325.
11. Slavicek R., Sato S. Bruxism – a function of the masticatory organ to cope with stress // *Wien. Med. Wochenschr*. 2004. No. 154. № 23-24. P. 584-589.

ГЕОПОЛИТИЧЕСКИЕ ПОЗИЦИИ РОССИИ В МИРОВОЙ ПОЛИТИКЕ XXI ВЕКА

Горшкова Е. А., Матвеев О. В.

*ФГБОУ ВО «Московский государственный
лингвистический университет», Москва,
e-mail: Genek1807@gmail.com,
matveev4you@yandex.ru*

В данной статье рассматривается эволюция геополитических позиций России, анализируются основные инструменты и направления их реализации, оцениваются ключевые вызовы и перспективы укрепления российского влияния в условиях формирующегося многополярного мира.

Актуальность обусловлена следующим:

1. России играет роль одного из главных архитекторов и «двигателей» перехода от однополярной модели к многополярной. Понимание доктринальных основ и практических инструментов российской стратегии необходимо для прогнозирования развития всей системы международных отношений.

2. Россия перестраивает систему своих внешних связей под воздействием жестких санкций и политической изоляции. Исследование практических шагов – от «поворота на Восток» и развития Севморпути до создания параллельных финансовых механизмов – позволяет оценить эффективность масштабной адаптации и ее долгосрочные последствия для глобальных рынков и альянсов, что крайне важно для государств и корпораций по всему миру.

3. Подчеркивая критическую зависимость геополитических амбиций от решения внутренних проблем (технологическая модернизация, преодоление сырьевой модели), работа задает реалистичные рамки для прогноза.

Наследие XX века и поиск новой идентичности

Распад Советского Союза в 1991 году создал принципиально новую геополитическую реальность для России. Потеря значительной части территории, населения и экономического потенциала, сокращение военного присутствия в мире, идеологический вакуум – все это поставило страну перед необходимостью переопределения своей роли в международных отношениях. 1990-е годы стали периодом адаптации к реалиям однополярного мира с доминированием США, попытками интеграции в западные институты и болезненными уступками в сфере национальных интересов.

Однако уже к концу 1990-х годов стало очевидно, что надежды на равноправное партнерство с Западом во многом не оправдались, чему свидетельствовали расширение НАТО на вос-

ток, игнорирование российских интересов в различных регионах мира. В российском политическом курсе началось формирование новой внешнеполитической концепции, основанной на принципах суверенитета, прагматизма и защите национальных интересов.

Формирование доктрины «суверенной демократии»

2000-е годы стали периодом становления новой геополитической философии России. Важнейшими вехами этого процесса стали:

1. Мюнхенская речь Владимира Владимировича Путина в 2007 году, в которой была подвергнута критике однополярная модель мира и заявлено о необходимости уважения суверенитета всех государств.

2. Разработка концепции «суверенной демократии», подчеркивающей право России на самостоятельный путь развития.

3. Формулировка принципов «русского мира» как цивилизационной общности, объединяющей народы исторической Руси.

В этот период Россия начинает более уверенно отстаивать свои позиции на международной арене, используя экономические инструменты (энергетическая дипломатия), военно-политические методы (модернизация армии) и дипломатические инициативы.

Современная концепция внешней политики: переход к активному формированию многополярного мира

Новая Концепция внешней политики Российской Федерации, утвержденная Указом Президента 31 марта 2023 года, представляет собой наиболее полное и системное изложение современных геополитических взглядов российского руководства. Документ констатирует завершение эпохи доминирования Запада и переход к формированию многополярной системы международных отношений [1].

Ключевые принципы, закрепленные в Концепции 2023 года:

1. Суверенное равенство государств, уважение их права выбирать модели развития, социального, политического и экономического устройства [1].

2. Неприятие гегемонии в международных делах [1].

3. Сотрудничество на основе баланса интересов и взаимной выгоды [1].

4. Невмешательство во внутренние дела [1].

5. Неделимость безопасности в глобальном и региональном аспектах [1].

6. Верховенство международного права при отказе от политики двойных стандартов [1].

7. Многообразие культур, цивилизаций и моделей организации общества [1].

Особое внимание в документе уделяется противодействию «гибридной войне нового типа» [1], которую ведут против России США и их союзники.

Эта концепция стала методологической основой для объяснения сложных процессов современной международной политики и обоснования мер по защите национального суверенитета.

ООН и международное право: борьба за легитимность

Несмотря на критику в адрес ООН и существующей системы международных отношений, Устав ООН, закрепляющий принципы суверенного равенства и невмешательства, остается для России центральной площадкой для легитимации своей внешней политики и защиты статуса великой державы. РФ придерживается позиции создания многополярного мира, в котором «ООН должна оставаться главной площадкой для прогрессивного развития и кодификации международного права» [1].

России выступает инициатором и лидером процесса адаптации международных институтов, в первую очередь ООН, к реалиям многополярного мира; настойчиво продвигает идею реформирования Совета Безопасности ООН с целью «восстановлению роли ООН в качестве центрального координирующего механизма в согласовании интересов государств – членов ООН и их действий по достижению целей Устава ООН» [1]. Это позволит легитимизировать многополярную архитектуру и повысить роль России как ответственного гаранта международной стабильности: «...прогрессивному развитию, в том числе с учетом реалий многополярного мира, и кодификации международного права, прежде всего в рамках усилий, предпринимаемых под эгидой ООН, а также обеспечению участия в международных договорах ООН наибольшего количества государств, единообразному толкованию и применению таких договоров» [1].

Экономические и энергетические инструменты геополитики

Экономические инструменты играют все более важную роль в реализации геополитических интересов России. В условиях санкционного давления происходит переориентация экономических связей с Западом на Восток и Юг. Ключевые направления этой политики:

1. Развитие Евразийского экономического союза (ЕАЭС) как центра интеграции на пространстве бывшего СССР.
2. Углубление экономического сотрудничества с Китаем в рамках стратегического партнерства.
3. Расширение торговых отношений со странами Южной Азии, Ближнего Востока, Африки и Латинской Америки.

4. Создание альтернативных финансовых механизмов (система МИР, СПФС) для снижения зависимости от долларовой системы.

Энергетическая дипломатия остается важным инструментом влияния, хотя ее эффективность снижается в условиях диверсификации поставок и развития альтернативных источников энергии. Тем не менее, Россия продолжает играть ключевую роль в обеспечении энергетической безопасности многих стран Европы и Азии.

Особое значение приобретает освоение Арктики и развитие Северного морского пути (СМП). Контроль над этим транспортным коридором не только укрепляет суверенитет России в арктическом регионе, но и позволяет предложить миру альтернативный логистический маршрут, сокращающий путь из Азии в Европу на 30-35% по сравнению с Суэцким каналом [2, с. 552].

Информационные и идеологические инструменты: формирование альтернативного нарратива

В условиях гибридной войны, как отмечено в Концепции внешней политики 2023 года, информационное противоборство становится одним из ключевых фронтов геополитического противостояния. Россия активно развивает собственные медиаресурсы на иностранных языках, инвестирует в публичную дипломатию.

Активно продвигается концепция «русского мира», который рассматривается как цивилизационная общность, обладающая свойством срединности и геополитической субъектности [3, с. 10]. Отмечается, что срединность России характеризуется свойством политического сознания, основанного на принципах управления большим пространством, заселенным народами, обладающими определенными социокультурными различиями [3, с. 11]. Это свойство является естественной предпосылкой формирования статуса одного из ключевых полюсов многополярного мира [3, с. 11]. Срединность оказывает влияние на формирование евразийской миссии России, способной обеспечить «гармоничный синтез, но не поглощение, всех культур народов, населяющих территорию на стыке Европы и Азии, Запада и Востока» [3, с. 10].

Поворот на Восток: стратегическое партнерство с Китаем и странами Азии

В условиях конфронтации с Западом «поворот на Восток» стал стратегическим императивом российской внешней политики. Ключевым партнером в этом направлении является Китай, отношения с которым развиваются в формате всеобъемлющего стратегического партнерства и взаимодействия.

Основные направления сотрудничества с Китаем:

1. Углубление торгово-экономических связей и увеличение товарооборота.

2. Сотрудничество в энергетической сфере (трубопроводы, СПГ).

3. Военно-техническое сотрудничество и совместные учения.

4. Координация позиций на международных площадках.

Помимо Китая, Россия развивает отношения с другими странами Азии, включая Индию, страны АСЕАН. Важную роль играет участие в Шанхайской организации сотрудничества (ШОС), которая рассматривается как платформа для взаимодействия в Центральной Азии.

БРИКС и ШОС: строительство альтернативной архитектуры

БРИКС и ШОС рассматриваются как важнейшие элементы формирующейся многополярной системы. Эти организации объединяют крупнейшие незападные экономики и играют все более важную роль.

В рамках БРИКС Россия продвигает идеи:

1. Создания альтернативных финансовых институтов (Новый банк развития).

2. Увеличения расчетов в национальных валютах.

3. Координации позиций по ключевым международным вопросам.

ШОС служит платформой для взаимодействия в сфере безопасности и экономического сотрудничества в Центральной Азии. Расширение организации (прием Ирана и Беларуси) свидетельствует о ее растущей привлекательности.

ЕАЭС: евразийский интеграционный проект

Евразийский экономический союз, созданный в 2015 году, представляет собой наиболее глубокую форму интеграции на постсоветском пространстве. Цель ЕАЭС – создание единого экономического пространства со свободным движением товаров, услуг, капитала и рабочей силы.

Для России ЕАЭС имеет не только экономическое, но и геополитическое значение:

1. Это инструмент укрепления влияния на постсоветском пространстве.

2. Платформа для взаимодействия с другими интеграционными объединениями.

«Глобальный Юг»: поиск новых союзников

В последние годы Россия активизировала взаимодействие со странами Африки, Латинской Америки. Эти регионы рассматриваются как потенциальные союзники в формировании многополярного мира. Основные инструменты работы с «глобальным Югом»:

1. Экономическое сотрудничество и инвестиции в инфраструктурные проекты.

2. Гуманитарная помощь и образовательные программы.

Кроме того, РФ инициирует и развивает диалоговые площадки, такие как Саммиты «Россия-Африка», «Россия-АСЕАН», что позволяет укреплять позиции в регионах «глобального Юга», где многие страны не поддерживают антироссийские санкции и заинтересованы в конструктивном сотрудничестве [4, с. 14-15].

Внешние вызовы: санкции, изоляция, конкуренция

Внешняя среда также создает серьезные трудности для реализации геополитической стратегии России:

1. Санкционное давление: введенные западными странами санкции затрагивают ключевые сектора экономики, ограничивают доступ к финансовым рынкам и технологиям, создают препятствия для международной торговли.

2. Дипломатическая изоляция: Россия столкнулась с беспрецедентной дипломатической изоляцией со стороны западных стран, что затрудняет ведение переговоров и продвижение своих инициатив на международных площадках.

3. Военно-политическое противостояние с НАТО: размещение дополнительных воинских контингентов и ударных вооружений НАТО в непосредственной близости от российских границ, а также регулярные масштабные учения расцениваются Москвой как прямая угроза национальной безопасности [5]. Стратегическая концепция НАТО 2022 года, объявившая Россию «прямой угрозой», направлена на ее сдерживание и ослабление всеми доступными средствами.

Внутренние вызовы: экономика, демография, технологии

Укрепление геополитических позиций России сталкивается с серьезными внутренними вызовами:

1. Экономическая зависимость от сырьевого экспорта: несмотря на некоторые успехи в импортозамещении, экономика России остается уязвимой к колебаниям мировых цен на сырьевые товары [6].

2. Технологическое отставание: санкции и ограничение доступа к передовым технологиям усугубляют отставание России в таких критически важных областях, как микроэлектроника, IT-технологии, биотехнологии. Недостаточные инвестиции в научные исследования и разработки затрудняют модернизацию экономики и укрепление технологического суверенитета [7].

3. Демографические проблемы: сокращение численности населения, старение нации, региональные диспропорции создают долгосрочные риски для экономического и оборонного потенциала страны [7].

Обобщение материала, раскрывающего геополитические позиции России в XXI веке, позволяет сформулировать некоторые уроки.

1. Способность России сформулировать и последовательно продвигать собственную концепцию мироустройства (от «суверенной демократии» и «русского мира» до «справедливой многополярности») стала ключевым инструментом легитимации своих действий на международной арене и привлечения союзников среди тех, кто не приемлет доминирование Запада. Урок: в современной гибридной конфронтации убедительная и последовательная идеологическая платформа является стратегическим активом, сравнимым с экономическими и военными ресурсами.

2. История российских геополитических позиций в XXI веке – это урок системной адаптации к меняющимся обстоятельствам. Столкнувшись с нарастающим противостоянием с Западом, Россия предприняла масштабный стратегический «поворот на Восток» и к «глобальному Югу». Диверсификация партнерств (Китай, Индия, страны Азии, Африки), создание параллельных финансовых механизмов и логистических коридоров (Севморпуть) показывают, как государство может использовать внешнее давление как стимул для перестройки своей системы связей и снижения уязвимости.

3. Статья четко указывает, что долгосрочная устойчивость геополитических позиций России зависит от решения проблем, не связанных напрямую с дипломатией или армией: преодоления сырьевой зависимости, технологического отставания и демографического кризиса.

Таким образом, результаты обобщаемого материала позволяют сделать следующие *выводы*:

1. Современная геополитическая позиция России заключается в идее строительства многополярного мира, основанного на принципах суверенитета, неделимой безопасности и цивилизационного многообразия. Этот курс институционализирован в Концепции внешней политики 2023 года и реализуется через альтернативные структуры (БРИКС+, ШОС, ЕАЭС).

2. Для России ООН остается ключевой легитимной площадкой для защиты суверенитета и продвижения идеи многополярного мира, особенно в условиях внешних вызовов – санкций и противостояния с НАТО.

3. Россия закрепила как один из центров силы, способный влиять на мировую политическую обстановку. Однако ее потенциал для роли «конструктора» устойчивой многополярности напрямую зависит от успеха в преодолении сырьевой модели экономики, технологической модернизации и решения демографических задач.

Список литературы

1. Указ Президента РФ от 31.03.2023 г. № 229 «Об утверждении Концепции внешней политики Российской Федерации» [Электронный ресурс]. URL: <http://kremlin.ru/acts/bank/49090> (дата обращения: 14.12.2025).

2. Юнфенг Ч. Геополитическое значение арктических судоходных путей и стратегия совместного освоения Китая и России // International scientific journal of Biruni. 2025. № 1. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/geopoliticheskoe-znachenie-arkticheskikh-sudohodnyh-putey-i-strategiya-sovmestnogo-osvoeniya-kitaya-i-rossii> (дата обращения: 14.12.2025).

3. Гребнев Р. Д. Теоретико-методологические аспекты роли и статуса центров геополитического влияния в многополярном мире // Вестник Московского университета. Серия 27. Глобалистика и геополитика. 2023. № 2. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/teoretiko-metodologicheskie-aspekty-rol-i-statusa-tsentrov-geopoliticheskogo-vliyaniya-v-mnogopolyarnom-mire> (дата обращения: 14.12.2025).

4. Ильин И. В. Формирующиеся тенденции глобальных политических процессов // Вестник Московского университета. Серия 27. Глобалистика и геополитика. 2023. № 3. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/formiruyuschiesya-tendentsii-globalnyh-politicheskikh-protsessov> (дата обращения: 14.12.2025).

5. Указ Президента РФ от 25 декабря 2014 г. № Пр-2976 «Военная доктрина Российской Федерации» [Электронный ресурс]. URL: https://www.mid.ru/ru/foreign_policy/official_documents/1584621/ (дата обращения: 14.12.2025).

6. Бурганова И. Н. Внешняя политика РФ: новые вызовы и угрозы // Архонт. 2020. № 1(16). [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/vneshnyaya-politika-rf-novye-vyzovy-i-ugrozy> (дата обращения: 14.12.2025).

7. Указ Президента РФ от 07.05.2018 г. № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года» (с изменениями и дополнениями) [Электронный ресурс]. URL: <http://kremlin.ru/acts/bank/43027> (дата обращения: 12.10.2025).

**РАЗРАБОТКА
ПРОГРАММНОГО КОМПЛЕКСА
ДЛЯ АВТОМАТИЗИРОВАННОЙ
ПРОВЕРКИ НОРМАЛЬНОСТИ
РАСПРЕДЕЛЕНИЯ И
СТАТИСТИЧЕСКОГО АНАЛИЗА
БИОМЕТРИЧЕСКИХ ДАННЫХ**

Ананченко И. В., Жижина К. П.,
Мельников И. Д.

ФГБОУ ВО «Санкт-Петербургский
государственный технологический институт
(технический университет)», Санкт-Петербург,
e-mail: anantchenko@yandex.ru,
ksen_pav@mail.ru, chghmelnikoff@yandex.ru

Введение

Проверка соответствия эмпирических данных нормальному распределению является обязательным этапом перед применением параметрических статистических методов в биологических исследованиях. Одним из классических способов такой проверки является критерий согласия χ^2 Пирсона. Его ручной расчет, включающий построение вариационных рядов, вычисление теоретических частот и сопоставление распределений, отличается высокой трудоемкостью и склонностью к ошибкам, особенно при работе с большими выборками данных.

Целью работы является разработка на языке Python с использованием библиотеки Tkinter специализированного программного обеспечения для автоматизации полного цикла статистического анализа биометрических данных с акцентом на проверку гипотезы о нормальности распределения по критерию Пирсона.

Актуальность разработки обусловлена повсеместной распространённостью нормального распределения в природе, следующей из центральной предельной теоремы. Поскольку многие параметрические критерии в биометрии требуют соответствия данных нормальному закону, автоматизация данного анализа становится практической необходимостью. Она позволяет ускорить обработку данных, минимизировать ошибки и сделать сложные статистические процедуры более доступными для исследователей в биологии и сельском хозяйстве.

**Математическое описание
и методы исследования**

По правилу Стерджеса число интервалов вариационного ряда K (ед.) рассчитывается по формуле [1]:

$$K = 1 + 3,322 \cdot \lg(n)$$

где n – объем выборки, ед.

Расчет: определяет оптимальное количество групп для группировки исходных данных.

Выборочное среднее значение признака \tilde{x} (г) определяется через соотношение [1]:

$$\tilde{x} = \frac{\sum_{i=1}^K x_i \cdot f_i}{n}$$

где x_i – середина i -го классового интервала, г; f_i – частота встречаемости вариант в i -м интервале, ед.

Расчет: характеризует центральную тенденцию распределения массы орехов в выборке.

Выборочная дисперсия s^2 (г²) и среднее квадратическое отклонение s (г) определяется по формуле [1]:

$$s^2 = \frac{\sum_{i=1}^K (x_i - \tilde{x})^2 \cdot f_i}{n - 1}$$

$$s = \sqrt{s^2}$$

Расчет: характеризуют вариацию (разброс) индивидуальных значений массы вокруг среднего.

Статистика критерия согласия χ^2 Пирсона рассчитывается по формуле [1]:

$$\chi^2_{\text{набл}} = \sum_{i=1}^K \frac{(f_i - F'_i)^2}{F'_i}$$

где $\chi^2_{\text{набл}}$ – наблюдаемое значение критерия Пирсона, безразмерная величина; f_i – эмпирическая (наблюдаемая) частота в i -м интервале, ед.; F'_i – теоретическая частота для i -го интервала, рассчитанная в предположении нормального распределения, ед.

Расчет: служит для количественной оценки расхождений между эмпирическим и теоретическим (нормальным) распределениями.

Доверительный интервал для генеральной средней μ (г) определяется по формуле [1]:

$$\tilde{x} - t_\alpha \cdot \frac{s}{\sqrt{n}} \leq \mu \leq \tilde{x} + t_\alpha \cdot \frac{s}{\sqrt{n}}$$

где t_α – критическое значение t -статистики (или z -статистики для больших n) для заданного уровня значимости α , безразмерная величина; s – выборочное среднее квадратическое отклонение, г.

Расчет: определяет диапазон, в котором с заданной доверительной вероятностью $P=1-\alpha$ находится истинное среднее значение массы орехов во всей генеральной совокупности.

Результаты исследования и их обсуждение

Разработан программный комплекс на Python с графическим интерфейсом Tkinter, автоматизирующий полный цикл статистического анализа.



Главное окно программы

Система выполняет проверку нормальности распределения по критерию χ^2 Пирсона и расчёт доверительных интервалов для параметров генеральной совокупности.

Интерфейс приложения представлен на рисунке. Он включает модуль импорта данных (CSV, Excel), настройку уровня значимости (α) и структурированный вывод результатов. Вычислительное ядро реализует построение вариационного ряда по правилу Стерджеса, расчёт выборочных характеристик, теоретических частот и статистики χ^2 , а также определяет доверительные интервалы для средней и стандартного отклонения.

Результаты представлены детально, включая промежуточные вычисления. Модуль визуализации на Matplotlib создаёт аналитические

графики: совмещённую гистограмму, график плотности нормального распределения и диаграмму доверительных интервалов.

Система стандартизирует процедуры проверки нормальности и оценки параметров, повышая воспроизводимость результатов и минимизируя вычислительные ошибки при обработке данных в исследовательских и прикладных задачах.

Исходный код проекта размещен в публичном репозитории на GitHub [<https://github.com/0Kopeika0/DEVELOPMENT-OF-A-SOFTWARE-PACKAGE-FOR-AUTOMATED-VERIFICATION-OF-THE-NORMALITY>].

Перспективным направлением является развитие системы в универсальную платформу статистического анализа. Это включает внедре-

ние альтернативных критериев нормальности (Шапиро-Уилка, Колмогорова-Смирнова) и методов для многомерных распределений. Практическую значимость усилит интеграция с системами сбора данных и внедрение устойчивых методов оценки. Переход к веб-архитектуре и облачным сервисам обеспечит широкую доступность и возможности для совместной работы. Создание специализированных модулей позволит применять систему в различных областях – от молекулярной биологии до промышленной стандартизации.

Выводы

Разработано программное обеспечение, автоматизирующее проверку нормальности распределения по критерию χ^2 Пирсона и оценку доверительных интервалов. Система с графическим интерфейсом Tkinter включает модули импорта данных, вычислительное ядро и визуализацию на Matplotlib. Её внедрение стандартизирует процедуры, сокращает время обработки, минимизирует ошибки и повышает доступность методов для специалистов. Перспективы развития связаны с добавлением новых критериев нормальности, интеграцией с системами сбора данных и переходом к веб-архитектуре для создания универсальной аналитической платформы.

Список литературы

1. Биометрия: учебник для вузов / П. С. Катмаков, В. П. Гавриленко, А. В. Бушов; под общей редакцией П. С. Катмакова. 3-е изд., перераб. и доп. М.: Юрайт, 2025. 186 с. (Высшее образование). ISBN 978-5-534-15732-1.
2. Федоров Д. Ю. Программирование на языке высокого уровня Python: учебное пособие. 2-е изд., пер. и доп. М.: Юрайт, 2019. 161 с. (Бакалавр. Прикладной курс). ISBN 978-5-534-10971-9. EDN WGENGH.

ВЛИЯНИЕ МУЛЬТИМЕДИА НА СОВРЕМЕННОЕ ОБРАЗОВАНИЕ

Антоненко А. С., Окулова Е. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: elenaokulova1987@yandex.ru

Научный руководитель: Окулова Е. А.

Введение

Влияние мультимедиа на современное образование произвело революцию в том, как студенты учатся и взаимодействуют с образовательными материалами. Мультимедиа, которое включает текст, изображения, аудио, видео и интерактивный контент, стало незаменимым инструментом в образовании, способствуя вовлечению, улучшая учебный процесс и делая образование более доступным для широкой аудитории. В этой статье мы рассмотрим, как мультимедиа влияет на образование, его преимущества и возникающие при этом проблемы.

Цель исследования: рассмотреть виды мультимедиа в современном образовании. Проанализировать внедрение таких технологий в учебный процесс. Выделить преимущества и недостатки данного направления. Найти варианты, как еще можно разнообразить образование с помощью мультимедийных технологий.

Материал и методы исследования

Материал и методы исследования: исследование влияния мультимедиа на образование включает в себе совокупность методов, например:

- анализ теоретических источников;
- наблюдение;
- анализ результатов деятельности;
- методы математической обработки;
- изучение и анализ психолого-педагогической и методической научной литературы.

Материалом исследования являются: мультимедийные средства обучения, мультимедийные тренажеры.

Результаты исследования и их обсуждение

1. Увеличение вовлеченности студентов через мультимедиа.

Влияние мультимедиа на современное образование особенно заметно в том, как оно увеличивает вовлеченность студентов. Традиционные методы обучения уже не справляются со своей задачей на 100%, которые часто полагаются исключительно на учебники и лекции, чаще всего они являются менее эффективными в захвате внимания студентов, особенно в цифровую эпоху. Мультимедиа вводит динамичные элементы, такие как видео, анимации и интерактивные симуляции, которые делают обучение более захватывающим и актуальным. С приходом технологий, а именно: небольшие видеоролики, которые меняются один за одним, листовая лента социальных сетей, или же игры, в которых меняется сюжетная линия каждую минуту и др. развивают так называемую “клиповую память”. Это все повлияло на восприятие информации с окружающего мира. Очень много обучающихся сталкиваются с проблемой фокусирования своего внимания на материале, который подается в виде монотонных рассказов или же просто прочтения материала с учебника.

Например, такие предметы, как физика, химия или история, могут оживать благодаря мультимедийным презентациям, которые позволяют студентам визуализировать сложные концепции или исторические события. Хорошо сделанное видео о историческом событии может создать более глубокую эмоциональную связь, чем простое чтение учебника. Более того, мультимедийные элементы, такие как викторины, интерактивные диаграммы и игровые обучающие активности, поддерживают мотивацию студен-

тов и их активное участие в учебном процессе. Также, такие методы помогают в игровой форме оценить знания обучающихся, не вызывая у него лишнего стресса.

2. Улучшение запоминания и понимания

Еще одно важное влияние мультимедиа на современное образование – это его способность улучшать запоминание и понимание. Мультимедиа позволяет преподавателям учить различные стили обучения, будь то визуальный, аудиальный или кинестетический. Например, визуальные учащиеся могут извлечь пользу из изображений, диаграмм и видео, в то время как аудиальные ученики могут предпочитать слушать подкасты или аудиолекции. Это помогает развивать все способы восприятия информации у обучающихся, а также, в некоторых случаях, очень хорошо развивает многозадачность.

Кроме того, мультимедиа помогает разбить сложные концепции на более усвояемые части. Например, анимации на уроках биологии могут помочь проиллюстрировать сложные процессы жизненно необходимых процессов организма человека, либо же природных процессов в понятной форме. Благодаря многократному взаимодействию с мультимедийным контентом, обучающиеся лучше запоминают информацию, которую они изучают, что приводит к улучшению академической успеваемости, а также качеству знаний. Можно проводить эксперименты не отходя от своего рабочего места, что очень удобно.

Каждый ученик по-разному усваивает информацию, поэтому подавать материал в разных видах или же хотя бы от разных спикеров – является необходимостью.

3. Доступность и инклюзивность

Влияние мультимедиа на современное образование выходит за рамки вовлеченности и запоминания, поскольку оно также способствует большей доступности и инклюзивности в образовании. Мультимедийные технологии сделали учебные материалы более доступными для студентов с ограниченными возможностями и теперь тяга к знаниям не имеет ограничений. Например, студенты с нарушениями слуха могут получить доступ к субтитрам или переводам на язык жестов, в то время как студенты с нарушением зрения могут использовать программы чтения с экрана или слушать аудиоконтент.

Более того, мультимедийные платформы предоставляют гибкие варианты обучения, позволяя студентам получать доступ к образовательным материалам в любое время и в любом месте. Онлайн-курсы и платформы электронного обучения позволяют студентам из различных географических местностей получать качественное образование, помогая сократить разрыв между привилегированными и обделенными учащимися.

4. Проблемы интеграции мультимедиа в образовании

Хотя влияние мультимедиа на современное образование в основном положительное, есть проблемы, связанные с его интеграцией. Одной из основных проблем является цифровое неравенство, когда студенты из недофинансируемых школ или бедных районов могут не иметь доступа к необходимым технологиям или интернет-соединениям, требуемым для полного использования образования на основе мультимедиа. Это создает неравенство в образовательных возможностях, поэтому эти нововведения не могут найти применения в таких учебных заведениях.

Кроме того, учителям необходимо надлежательное обучение, чтобы эффективно интегрировать мультимедиа в свои методы преподавания. Хотя мультимедиа предлагает множество преимуществ, оно может быть неэффективным или даже контрпродуктивным, если используется неправильно. Преподаватели должны знать, как курировать контент, управлять временем нахождения перед экраном и балансировать традиционные методы обучения с цифровыми инструментами. Полностью отказаться от лекций и учебников, на данный момент, невозможно.

5. Будущее мультимедиа в образовании

Смотря в будущее, ожидается, что влияние мультимедиа на современное образование продолжит расти. С учетом достижений в технологиях, погружающие учебные опыты, такие как виртуальная реальность (VR) и дополненная реальность (AR), становятся все более распространенными. Эти инструменты предлагают беспрецедентные возможности для практического обучения, позволяя студентам исследовать виртуальные миры, проводить виртуальные научные эксперименты или посещать исторические места, не покидая класс.

Кроме того, искусственный интеллект (AI) трансформирует способ доставки образовательного контента. Платформы на основе ИИ могут персонализировать мультимедийный контент в зависимости от индивидуальных стилей и потребностей учеников, предлагая адаптированные учебные опыты, которые намного более эффективны, чем универсальный подход.

Заключение

В заключение, влияние мультимедиа на обучение невозможно полноценно оценить, предлагая множество преимуществ, таких как увеличение вовлеченности, улучшение запоминания и повышение доступности. Однако необходимо решить такие проблемы, как цифровое неравенство и необходимость подготовки учителей, чтобы все студенты могли воспользоваться этими технологиями. По мере развития мультимедиа его роль в формировании будущего образова-

ния будет только расти, делая обучение более интерактивным, инклюзивным и эффективным для всех.

Список литературы

1. Новые педагогические и информационные технологии в системе образования. Учеб. кассовый терминал для студ. пед. вуз и система POV повыш. квалиф. пед. кадров / под ред. А. И. Татаркина. Е. С. Полат. М.: Издательский дом «Академия», 2017.
2. Maktabgacha ta'limda yangi pedagogik va axborot texnologiyalarini tatbiq etishning dolzarb masalalari. Respublika ilmiy-amaliy konferentsiya materiallari. 2010 yil 5–6 may. Qarshi, «Qarshi Davlat universiteti» nashriyoti, 2010.
3. Renee Hobbs. Digital and Media Literacy: A Plan of Action. Washington, D.C., 2010.
4. Абылова Г. Ж. Роль и место мультимедиа в образовании, 2025. URL: <https://cyberleninka.ru/article/n/rol-i-mesto-multimedia-v-obrazovanii> (дата обращения: 15.12.2025).
5. Иванова Н.В. Эффективность применения мультимедийных технологий при обучении в высшей школе, 2020. URL: <https://www.dissercat.com/content/effektivnost-primeneniya-multimediyuykh-tehnologiy-pri-obuchanii-v-vyshej-shkole> (дата обращения: 15.12.2025).

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ТУРИСТИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ

Бабенко Н. Н.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: oksana.strokan0@mail.ru*

Научный руководитель: Строкань О. В.

Введение

Современный туристический рынок характеризуется высокой конкуренцией. Успех туристического агентства во многом зависит от способности привлекать и удерживать клиентов. Клиентоориентированный подход требует знания потребностей и предпочтений клиентов, что невозможно без систематизированной информации о них [2]. Разработка эффективных маркетинговых кампаний требует сегментации клиентской базы и персонализации предложений. В последнее время все больше руководителей начинают четко осознавать важность построения в организации информационной системы как необходимого инструментария для успешного управления бизнесом в современных условиях. Благодаря развитию сетевых технологий локальные информационные системы повсеместно вытесняются клиент-серверными и многоуровневыми реализациями. В связи с выше сказанным, использование информационных систем туристического агентства является актуальной и стратегически важной задачей, направленной на повышение конкурентоспособности, улучшение качества обслуживания, оптимизацию бизнес-процессов и обеспечение соответствия требованиям законодательства.

Цель исследования – на основе анализа предметной области предложить современную информационную систему для использования в сфере туристической деятельности.

Материал и методы исследования

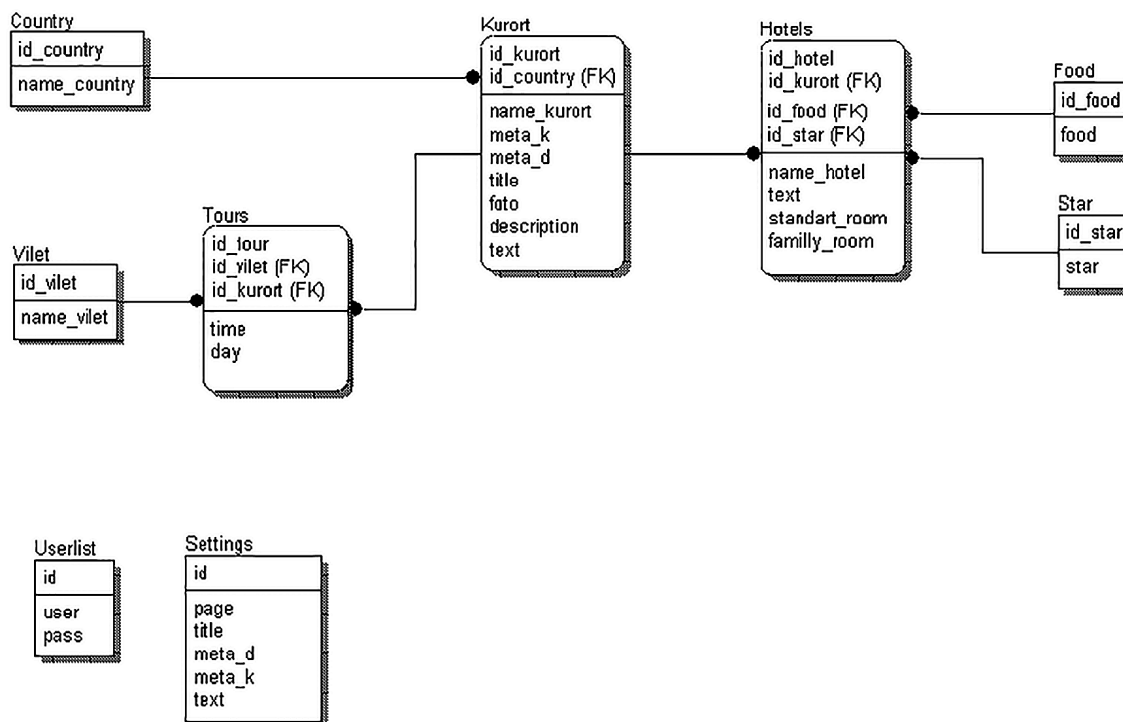
Актуальность использования информационной системы для туристического агентства сегодня обусловлена тем, что использование современных информационных систем позволяет решить ключевые проблемы бизнеса, улучшить качество обслуживания клиентов, повысить конкурентоспособность и адаптироваться к меняющимся требованиям рынка [2, 8].

Результаты исследования и их обсуждение

Сегодня индустрия туризма активно трансформируется благодаря новым информационным системам и технологиям. Автоматизация делает туристический бизнес глобальным и компьютеризированным, объединяя авиакомпании, гостиничные сети и туристические корпорации по всему миру. Современные ИТ-решения позволяют сделать туристические услуги более доступными и гибкими для каждого клиента [2, 8].

На рынке программного обеспечения предлагаются различной функциональности и масштабыности информационные системы для оптимизации работы туристических предприятий любых размеров. Среди наиболее популярных информационных систем являются большие веб-сайты со списком предлагаемых услуг, галерей изображений, отзывами клиентов, контактными данными и т.д. [3-5] Особенностью таких информационных систем является обязательное наличие форм связей и баз данных, где, собственно, хранится вся информация о клиентах и услугах. Результаты анализа показывают, что большинство онлайн-турагентств страдают от схожих проблем, в основном связанных с неудобным интерфейсом и неэффективным представлением информации. Кроме того, многие из них не используют омниканальность для взаимодействия с клиентами. В связи с полученными результатами, в данной работе предлагается информационная система туристического агентства в виде веб-сайта, которая позволяет автоматизировать процесс бронирования авиабилетов, отелей и других услуг, значительно сокращая время, затрачиваемое на оформление тура.

Для функционирования клиентской базы информационной системы туристического агентства, а также для реализации запросов, необходимо создать базу данных, в таблицах которой хранится информация о предлагаемых для отдыха странах, курортах и отелях. На рисунке представлена логическая модель базы данных.



Логическая модель базы данных

Данные, представленные на рисунке, содержат следующую информацию:

- «Settings»: содержит информацию по каждой странице системы;
- «Userlist»: содержит имя и пароль пользователей для доступа в администраторскую часть;
- «Country»: таблица с названиями всех стран, по которым работает туристическое агентство;
- «Kurort»: содержит информацию о каждом курорте в странах, с которыми работает туристическое агентство;
- «Hotels»: содержит информацию об отелях на каждом курорте;
- «Star»: содержит информацию о звездной категории отеля;
- «Food»: содержит информацию о типе питания в отеле;
- «Vilet»: содержит названия городов, из которых осуществляется вылет на курорты.
- «Tours»: соединительная таблица, в которой находится информация о городе вылета.

Независимо от размеров базы данных или объема данных в ней, вопрос эффективного управления и обеспечения стабильного доступа к данным требует от разработчиков использования специальных структурированных запросов для работы с базами данных. Для управления данными используются СУБД, а также специализированный язык для формирования запросов. Для выполнения поставленных задач в рамках данной работы, была выбрана СУБД MySQL. Работать с MySQL можно как в текстовом режиме с помощью написания специальных запросов

на языке PHP, так и в графическом. Например, с помощью визуального интерфейса для работы phpMyAdmin, который позволяет значительно упростить работу с базами данных в MySQL.

MySQL – это реляционная система управления базами данных с открытым исходным кодом. База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми – от простого списка будущих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, которым и является программное обеспечение MySQL. Поскольку компьютеры прекрасно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному – как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений [3]. Главным фактором языка PHP является практичность. PHP предоставляет программисту средства для быстрого и эффективного решения поставленных задач. Практический характер PHP обусловлен пятью важными характеристиками: традиционностью; простотой; эффективностью; безопасностью; гибкостью [6].

Данная модель базы данных является основой для информационной системы для использования в сфере туристической деятельности.

Заключение

В ходе проведенных исследований выявлена необходимость использования современных информационных систем в сфере туристической деятельности. В связи со спецификой туристического бизнеса, развитие информационной инфраструктуры является важным фактором конкурентоспособности туристических агентств. В этом контексте, разработанная структура базы данных информационной системы туристического агентства является важным шагом в обеспечении эффективности и качества обслуживания клиентов, а также в увеличении производительности и оптимизации деятельности предприятия в целом. Данное решение позволяет повысить эффективность работы, улучшить качество обслуживания клиентов, увеличить продажи и прибыльность, получить конкурентные преимущества и улучшить управление и контроль за деятельностью агентства.

Список литературы

1. Диго С. М. Базы данных: проектирование и использование: учебное пособие для вузов. М.: Финансы и статистика, 2005. 87 с.
2. Коноплева В. С., Герасимова А. В. Туристский бизнес: проблемы и пути решения // Экономический вектор. 2023. №2 (33).
3. Романенко Д. М., Осоко С. А. Программирование на языке PHP: учеб.-метод. пособие для студентов специальности 1-47 01 02 «Дизайн электронных и веб-изданий». Минск: БГТУ, 2021. 256 с.
4. Сайт онлайн-турагентства «Onlinetours» // Sputnik. URL: <https://www.onlinetours.ru/> (дата обращения: 15.09.2025).
5. Сайт онлайн-турагентства «ОнлайнТур» // ОнлайнТур. URL: <https://www.onlinetur.ru/> (дата обращения: 16.09.2025).
6. Сайт онлайн-сервиса для организации путешествий «OneTwoTrip» // ООО «Вайт Тревел». URL: <https://www.onetwotrip.com/ru/> (дата обращения: 15.12.2025).
7. Федеральный закон «Об основах туристской деятельности в Российской Федерации» от 24.11.1996 г. № 132-ФЗ. М.: Эксмо, 2019. 48 с.
8. MySQL. URL: <https://www.mysql.com.> (дата обращения: 01.10.2025).

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ РЕЧИ: СОСТОЯНИЕ И НАПРАВЛЕНИЯ РАЗВИТИЯ

Бахарев Б.А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: anna.dyachenko597@mail.ru*

Научный руководитель: Дяченко А.С.

Введение

Распознавание речи является одной из ключевых технологий взаимодействия человека с цифровыми продуктами. Голосовые интерфейсы активно используются в мобильных приложениях, интеллектуальных ассистентах, системах «умного дома» и сервисах поддержки пользователей [1]. По сравнению с традиционным текстовым вводом голосовое управление обла-

дает более естественным и интуитивным характером, что делает его перспективным направлением развития пользовательских интерфейсов.

Однако, несмотря на значительный прогресс, современные системы распознавания речи всё ещё сталкиваются с рядом ограничений. Пользователю часто приходится подстраиваться под заранее заданные команды, из-за чего диалог становится менее естественным. К основным проблемам относятся:

1. Недостаточная гибкость диалогового взаимодействия, при которой система плохо обрабатывает свободную, естественную речь пользователя.
2. Ограниченность словарного запаса и языковых моделей, особенно в случае использования профессиональной или специализированной лексики.
3. Сложности распознавания речи в условиях фонового шума, а также при наличии акцентов, диалектных особенностей и индивидуальных особенностей произношения.

Цель исследования – проанализировать проблемы и перспективы развития технологий распознавания речи, а также оценить их влияние на качество пользовательского взаимодействия с цифровыми системами.

Материалы и методы исследования

В ходе исследования использовались методы анализа и обобщения научных публикаций, сравнительного анализа существующих технологий распознавания речи, а также изучение практики их применения в различных отраслях [2]. Эмпирической базой послужили данные о внедрении голосовых технологий в мобильных приложениях, колл-центрах, медицинских и интеллектуальных информационных системах.

Результаты исследования и их обсуждение

Голосовой ввод в современных цифровых продуктах становится одним из ключевых способов взаимодействия пользователя с системой. В ряде приложений, таких как интеллектуальные системы управления и сервисы умного дома, он выполняет основную функциональную роль, обеспечивая удобство и скорость доступа к возможностям цифровой среды [1].

В мобильных и веб-приложениях используются различные формы голосового ввода, отличающиеся по степени интеграции и функциональности. Основные типы представлены в таблице.

Функционирование систем распознавания речи, как правило, основано на сочетании языковой и акустической моделей. Языковая модель отвечает за прогнозирование последовательности слов, тогда как акустическая модель сопоставляет звуковой сигнал с вероятными фонемами [2]. Связующим элементом между ними выступает словарь, содержащий информацию о произношении и базовых грамматических правилах.

Основные типы голосового ввода в цифровых приложениях

Тип голосового ввода	Краткая характеристика
Голосовой ввод через клавиатуру	Преобразует речь пользователя в текст для ввода в поисковые строки и мессенджеры; требует высокой точности распознавания
Встроенные голосовые ассистенты	Интегрированы в операционную систему; работают на основе ограниченного набора команд
Голосовой ввод в браузерах и сервисах	Используется совместно с технологиями искусственного интеллекта, обеспечивая более гибкий диалог



Этапы развития технологий распознавания речи

Несмотря на технологический прогресс, точность распознавания снижается при работе с узкоспециализированной лексикой (медицинской, технической, юридической), а также в условиях фонового шума, перекрытия голосов и индивидуальных речевых особенностей пользователей. Эти факторы остаются ключевыми ограничениями для широкого и универсального применения голосовых интерфейсов.

Качественный этап развития технологий распознавания речи связан с внедрением методов глубокого обучения и искусственного интеллекта (рисунок). Современные нейросетевые модели учитывают не только звучание слов, но и контекст высказывания, что делает голосовое взаимодействие более естественным и гибким по сравнению с ранними системами, ограниченными набором команд [2].

Несмотря на значительный прогресс, сохраняются нерешённые проблемы. Крупные технологические компании добились высоких результатов в массовых голосовых сервисах, однако в специализированных областях – медицине, инженерии и юриспруденции – точность распознавания остаётся низкой. Основной причиной является недостаточная поддержка профессиональной лексики, что снижает качество пользовательского опыта и доверие к технологии [3].

Важным фактором эффективности голосового ввода остаётся безопасность данных. Пото-

ковая передача голосовой информации повышает риски утечек, поэтому всё чаще используется локальная обработка, требующая дополнительных вычислительных ресурсов и способная влиять на производительность систем [3, 4].

Современные платформы применяют анализ эмоций на основе текстового содержания и аудиопараметров речи, что позволяет более точно оценивать состояние пользователя, однако данный подход ограничен индивидуальными и культурными особенностями речи.

Наиболее широко технологии распознавания речи используются в колл-центрах для анализа обращений, повышения качества обслуживания и обучения персонала [1]. Аналогичные решения применяются в здравоохранении – для ведения документации, поддержки пациентов и мониторинга состояния здоровья. Технологии также находят применение в сфере безопасности и правоохранительной деятельности для анализа больших массивов аудиоданных, что требует строгого нормативного регулирования.

Одним из ключевых направлений развития является аналитика речи в реальном времени, при которой обработка аудиоданных осуществляется непосредственно в ходе диалога. Это позволяет оперативно получать информацию о содержании разговора и реакции пользователя, повышая эффективность голосовых сервисов.

Другой значимой тенденцией является интеграция генеративного искусственного интеллекта в системы распознавания речи. Большие языковые модели и технологии синтеза речи обеспечивают создание виртуальных ассистентов и голосовых ботов, способных к более естественному диалогу и уже применяемых в службах поддержки и цифровых помощниках [5].

Заключение

Проведённый анализ показал, что технологии распознавания речи эволюционировали от простых акустических моделей к интеллектуальным системам, учитывающим контекст и намерения пользователя. При этом сохраняются проблемы, связанные с точностью распознавания, обработкой специализированной лексики и интерпретацией эмоциональных характеристик речи.

Дальнейшее развитие голосовых технологий обусловлено углублением интеграции искусственного интеллекта и расширением сфер их применения, при обязательном учёте пользовательского опыта, этических требований и профессиональной специфики.

Список литературы

1. Шматков В. Н., Бонковски П., Медведев Д. С., Корзунин С. В., Голендухин Д. В., Спыну С. Ф., Муромцев Д. И. Взаимодействие с устройствами интернета вещей с использованием голосового интерфейса // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19, № 4. С. 714–721. DOI: 10.17586/2226-1494-2019-19-4-714-721.
2. Баляба Я. В., Рычка О. В. Разработка приложения для распознавания речи // Информатика и кибернетика. 2023. № 1 (31). С. 5–11. EDN CMCNTY.
3. Нурмухаметов Р. К., Торин С. С. Цифровое доверие (digital trust): сущность и меры по его повышению // Известия Тульского государственного университета. Экономические и юридические науки. 2020. № 1. С. 32–39. EDN IDNRCS.
4. Шерман В. А. Анализ существующих систем управления содержанием // Студенческий научный форум: материалы XVIII Международной студенческой научной конференции (Москва, 20 октября 2024 г. – 15 марта 2025 г.). М.: ЕНПП, 2025. С. 117–119. EDN MOGAVJ.
5. Чепурной М. П., Дяченко А. С. Сравнительный анализ фреймворков для фронтенд-разработки // Международный студенческий научный вестник. 2025. № 1. С. 12. EDN COGVEA.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НАТИВНОЙ И КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Гринин А. О.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: anna.dyachenko597@mail.ru

Научный руководитель: Дяченко А.С.

В современной индустрии мобильных приложений существуют разные программные платформы, но на рынке уже долгое время

остаются две доминирующие операционные системы – Android и iOS. Наличие нескольких операционных сред ставит перед инженерами проблему выбора оптимальной технологической стека. На сегодняшний день сформировались два основных вектора разработки мобильных приложений. Первый – это классический нативный подход, требующий создания и поддержки уникальной кодовой базы для каждой операционной системы. Второй – кроссплатформенный, позволяющий использовать единый программный код для разных платформ. Существенные различия в архитектуре, стоимости и производительности этих методов обуславливают необходимость их детального изучения и сравнительного анализа [1].

Материалы и методы исследования

Актуальные научные работы в области мобильной разработки демонстрируют смещение исследовательского фокуса с чисто технических параметров производительности на комплексную оценку жизненного цикла программного продукта. Согласно выводам авторов [1], нативный подход остается приоритетным для проектов, требующих низкоуровневого управления ресурсами, в то время как кроссплатформенные решения позволяют значительно оптимизировать процесс разработки. Современные фреймворки, такие как Flutter и React Native, рассматриваются как эффективный инструмент, предлагающий баланс между кроссплатформенной совместимостью и скоростью вывода продукта на рынок [2].

Целью исследования является проведение сравнительного анализа нативного и кроссплатформенного подходов для определения наиболее эффективной стратегии разработки мобильных приложений в условиях дуополии платформ Android и iOS, учитывая баланс между качеством продукта и затратами на его производство.

Результаты исследования и их обсуждение

В современную технологическую эпоху смартфоны стали наиболее распространенными электронными устройствами, интегрированными практически во все сферы человеческой деятельности. Современный смартфон характеризуется не только его базовыми коммуникационными возможностями (звонками и сообщениями), но и расширенным функционалом: наличием систем глобального позиционирования (GPS), модулей биометрической аутентификации, высокотехнологичных камер и развитых операционных систем, обеспечивающих работу сложного программного обеспечения. Согласно статистическим данным на 2025 год, доминирующими платформами остаются Android с долей рынка 71,9% и iOS, занимающая 27,67% [3].

Сравнительный анализ подходов к разработке мобильных приложений

Критерий	Нативный подход	Кроссплатформенный подход
Языки программирования	Swift, Objective-C (iOS); Kotlin, Java (Android)	Dart (Flutter), JavaScript (React Native), C# (.NET MAUI), Kotlin (KMP)
Производительность	Прямой доступ к ресурсам процессора и GPU	Затраты ресурсов на слой абстракции
Доступ к API и железу	Полный доступ ко всем датчикам и системным функциям	Поддержка новых функций ОС внедряется с задержкой
Пользовательский интерфейс	Идеально плавный	Высокая схожесть, но возможны трудности с кастомной анимацией и специфическими жестами
Сложность поддержки	Необходимо обновлять и тестировать два независимых приложения	Изменения вносятся в один проект и применяются ко всем платформам сразу
Объем приложения	Минимальный (нет лишних библиотек и движков)	Значительный (из-за встроенного веса кроссплатформенного движка)
Безопасность	Максимальная за счет встроенных средств защиты конкретной ОС	Зависит от безопасности стороннего фреймворка
Целевой сегмент	Высоконагруженные системы, приложения со сложной графикой	MVP, корпоративные приложения

При наличии двух основных операционных систем имеет смысл ориентироваться на обе платформы при разработке приложения, предназначенного практически для всех пользователей.

Существуют два основных подхода к созданию мобильных приложений: нативная и кроссплатформенная. Нативная разработка подразумевает использование языков и инструментов, которые созданы непосредственно для конкретной операционной системы. Так, для Android это Java или Kotlin, а для iOS – Objective-C или Swift. Этот подход позволяет приложению напрямую обращаться к функциям смартфона, обеспечивая максимальную скорость работы, но требует написания отдельного кода для каждой платформы. В случае кроссплатформенного решения разработчики используют единый исходный код, который запускается на разных системах благодаря специальному промежуточному слою – абстракции. Этот слой переводит универсальные команды в понятный для конкретной ОС вид, что значительно экономит ресурсы при разработке. Среди наиболее известных инструментов такого типа выделяются: React Native, Flutter, .NET MAUI и Kotlin Multiplatform [4]. Подробное сравнение рассматриваемых подходов представлено в таблице.

Для масштабных и долгосрочных проектов приоритетным остаётся нативный подход к разработке. Он обеспечивает полное соответствие требованиям конкретной платформы, доступ к её аппаратным ресурсам и специализированным API, а также высокую производительность и качество пользовательского опыта. Приложения со сложной логикой, интерфейсом и анимацией требуют стабильности и масштабируемости, которые в полной мере реализуются

именно в нативной разработке [5]. Опыт крупных компаний показывает, что попытки перехода на кроссплатформенные решения нередко заканчиваются возвратом к нативным технологиям из-за ограничений по производительности и сложности масштабирования.

Кроссплатформенные технологии целесообразны для создания прототипов, пилотных решений и простых приложений со стандартизированным функционалом, где скорость разработки и снижение затрат являются приоритетом. Согласно данным Statista, около трети мобильных разработчиков используют кроссплатформенные фреймворки, однако большинство специалистов по-прежнему отдают предпочтение нативным технологиям, особенно при реализации сложных и ресурсоёмких проектов [6]. Таким образом, анализ существующих практик демонстрирует, что выбор подхода должен определяться не модными тенденциями, а целями проекта, его масштабом и требованиями к качеству конечного продукта, что создаёт основу для дальнейших обобщений и выводов.

Заключение

Проведённый анализ показал, что выбор между нативной и кроссплатформенной разработкой зависит от целей проекта. Нативный подход обеспечивает максимальную производительность, безопасность и качество пользовательского опыта и оптимален для ресурсоёмких и масштабируемых приложений. Кроссплатформенная разработка эффективна для быстрого запуска и снижения затрат, но имеет ограничения по производительности. Оптимальный выбор определяется балансом между качеством и скоростью разработки.

Список литературы

1. Кодитувака С., Пасиллоб Н. Разработка мобильного программного обеспечения в цифровую эпоху: сравнительная оценка кроссплатформенных фреймворков // JPO. 2025. Т. 8, № 2. С. 9–17.
2. Чепурной М. П., Дяченко А. С. Сравнительный анализ фреймворков для фронтенд-разработки // Международный студенческий научный вестник. 2025. № 1. С. 12. EDN COGVEA.
3. Mobile Operating System Market Share Worldwide // Statcounter Global Stats. [Электронный ресурс]. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата обращения: 25.12.2025).
4. Kotlin Multiplatform for cross-platform mobile development // Kotlin Documentation. [Электронный ресурс]. URL: <https://kotlinlang.org/docs/multiplatform/cross-platform-mobile-development.html> (дата обращения: 27.12.2025).
5. Дяченко А. С. Использование динамических элементов в дизайне сайтов для улучшения пользовательского опыта // Материалы XXV Всероссийской научно-практической конференции молодых ученых, аспирантов и студентов с международным участием, посвященной 80-летию Победы в Великой Отечественной войне 1941–1945 гг. (Нерюнгри, 23–25 октября 2025 г.). Якутск: Северо-Восточный федеральный университет им. М. К. Аммосова, 2025. С. 352–357. EDN OKPJJK.
6. Statista. Worldwide software developer working hours [Электронный ресурс]. URL: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/> (дата обращения: 26.12.2025).

**АСПЕКТЫ ИСПОЛЬЗОВАНИЯ
МУЛЬТИМЕДИЙНЫХ
ТЕХНОЛОГИЙ В ИСКУССТВЕ**

Гриценко М. С., Окулова Е. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: mmaria.gritsenko@yandex.ru*

Научный руководитель: Окулова Е. А.

Введение

В современном мире культурное пространство характеризуется неразрывным слиянием искусства и технологий, что дает повод для образования новейших форм художественного выражения. Мультимедиа становится частью художественного прогресса и включает в себя цифровые среды, программные платформы, НТТР. Классические формы искусства основаны на линейных и статичных носителях, например, живопись, скульптура, традиционный кинематограф. А мультимедийные технологии обладают динамикой и интерактивной природой, что кардинально меняет взаимосвязь между автором и зрителем. Актуальность нашего исследования обусловлена бурным развитием технологий и их внедрением в сферу искусства. В последние годы произошел определенный прорыв в области программного обеспечения и инфраструктуры интернет, что позволило авторам и художникам создавать новые формы художественного выражения. В сетевом искусстве применение интернета как платформы и поиска материала, компьютерная анимация, достигшая невероятного уровня реалистичности, а также

передовые технологии видеомонтажа – все это создает новую художественную эпоху, в которой размываются границы между дисциплинами и медиа. В связи с этим необходимо провести исследование применения мультимедийных технологий в искусстве, которые затрагивают не только технические, так и эстетические, философские и культурные измерения. Исключительно качественный подход даст описать не только современные тенденции, а и спрогнозировать будущее развитие цифровых художественных практик.

**Анализ последних исследований
и публикаций**

Исследования цифрового искусства чаще опираются на то, как технологии меняют художественные формы. Зачастую упуская при этом из виду более глубокий смысл между понятием об авторстве технической части, самой эстетики в целом. В публикациях об net art рассматривается критика цифровой среды. Многие статьи описывают яркие проекты, при этом нет объяснения как такие формы влияют на эстетическую и даже политические возможности сетевого искусства. В сфере анимации отечественные исследователи акцентируют внимание на коммерции. И редко затрагивают тему как игровые движки и программы с открытым доступом влияют на новое восприятие анимации в целом. Что касается видеомонтажа, не смотря на активные программные обеспечения крайне мало уделяется внимание созданию реалистичной графики и анимации. И все чаще приходим к коротким горизонтальным видео, которые построены не на основах кинематографа. Одним из сложных вопросов остается искусственный интеллект. Нейросеть участвует в создании картин, дизайна и музыки. Известные художники применяют инструменты ИИ, а известные дизайнеры экспериментируют. Крупные компании заключают договоры для расширения возможностей, при этом сохраняя свои права на уникальность. При всем этом нет четкого ответа об авторстве.

Цель исследования – системный анализ аспектов использования мультимедийных технологий в современном искусстве на примере четырех ключевых направлений: сетевого искусства, компьютерной анимации, видеомонтажа, актуальное развитие искусства и нейросети. Особое внимание уделяется трансформации художественных форм, эволюции зрительского восприятия и влиянию цифровой среды на креативные процессы.

Материал и методы исследования

В ходе исследования использовались следующие методы:

- изучение фундаментальных трудов по теории медиа, цифровой эстетике и современным

художественным практикам, опубликованных в научных журналах, индексируемых в базах данных;

- сопоставление традиционных и цифровых художественных форм с целью выявления специфики мультимедийных практик;

- детальный разбор конкретных произведений из области сетевого искусства компьютерной анимации (российские студии, такие как «Voronezh Animation Studio», «Союзмультфильм»), а также примеров инновационного видеомонтажа современного кино.

Материалом исследования послужили научные публикации российских и зарубежных авторов за период с 2015 по 2025 гг., а также примеры художественных произведений, представленные на международных площадках цифрового искусства

Результаты исследования и их обсуждение

1. *Сетевое искусство.* Сетевое искусство фактически является одним из кардинальных проявлений мирового искусства и цифровой среды. Его суть заключается в том, что ключевая аудитория существует в интернете. В отличие от музеев, галерей и выставок, сетевое искусство имеет общественную ориентацию, подчеркивает полноту охвата, принадлежность всем без исключения. В Российской Федерации признают, что net art занимает определенное место в цифровой культуре. Детальный анализ доказывает, что свойство сети зависит от протоколов, алгоритмов и платформ, что делает его крайне уязвимым. Высокая уязвимость дает повод для исследования, анализа и методических основ и акцентирует причинно-следственную связь: технологии так или иначе определяют социальные изменения. Интересными результатами использования net art в искусстве достигаются подходом реди-мейд, проще говоря применение предмета не по назначению: когда художники создают неожиданные сборки, помещают предмет в непривычный для него контекст, пытаются перевернуть повседневную ситуацию. В пример можно привести деятельность швейцарского дуэта Mediengruppe Bitnik, чьи остроумные проекты часто граничат с нарушением законов. Иск о нарушении авторских прав могла обернуться их акция 2007 года «Вызывает опера. Арии для всех». Художники спрятали в зале Цюрихского оперного театра «жучки» и транслировали постановки на телефонные номера жителей в городе Цюрих, выбранные случайным образом. Приняв звонок, местный житель получал возможность слушать прямую трансляцию оперного спектакля (1).

2. *Компьютерная анимация: от инструмента к художественной среде.* Анимация давно утвердилась как одно из самых универ-

сальных искусств – она преодолевает языковые, культурные и религиозные барьеры, оставаясь понятной зрителям по всему миру. Хотя интерес к российской анимации всегда присутствовал, долгое время она оставалась в тени на международной арене. Лишь отдельные работы, участвовавшие в фестивалях или попадавшие в эфир глобальных платформ, находили зарубежную аудиторию. Ситуация начала меняться в последнее десятилетие: российские анимационные студии активно осваивают международные рынки. Проекты вроде «Смешариков», «Маши и Медведя», «Фиксиков», «Тимы и Томы», «Алисы знает, что делать!» и «Трёх котиков» не только завоевали популярность у отечественной публики, но и получили широкое распространение за рубежом. Особенно выделяется мультсериал «Маша и Медведь». На сегодняшний день шоу локализовано на 37 языков и транслируется в более чем 150 странах, став ярчайшим примером того, как российская анимация может добиться глобального признания [2]. На данный момент анимация переживает этап трансформации по сей день: из дополнительного элемента кинопроизводства она превратилась в отдельную художественную среду, в которой соединяют инновационные программные и производственные технологии. Переход присутствует в работе ведущих анимационных студий, таких как Wizart Animation (с 2024г.- Voronezh Animation Studio) и «Союзмультфильм», которые динамично осваивают продвинутые 3D-технологии, технологии захвата движения и комплексы программ для гейм-разработки. Главным технологическим сдвигом стало применение игровых движков, таких как Unreal Engine, для разработки анимационного контента. Такие программы позволяют реализовать рендинг в реальном времени, существенно сокращая сроки визуализации данных. Союзмультфильм в сотрудничестве с Magic Factory Animation применяют Unreal Engine 4 и Unreal Engine 5 создания анимации где преобладает высшая детализация визуальной среды [3]. Что позволяет сократить рабочее время, гибко управлять текстурами камерой без повторной генерации изображения. Особое внимание уделяется технологиям захвата движения (motion capture). Хотя полный оптический захват пока экономически нецелесообразен для большинства российских студий, Союзмультфильм применяет гибридные методы: на камеру записывается актерская игра, а ключевые позы вручную переносятся в 3D-модель [5]. Студия Voronezh Animation Studio, использует промышленные процесс на базе Autodesk Maya, ZBrush, Substance Painter и RenderMan, Blender, что обеспечивает высокое качество моделирования персонажей, текстурирование и финальной визуализации. 3D-редактор

Blender является бесплатным программным обеспечением, поддерживается платформами Windows, Linux, Mac OS X, FreeBSD и распространяемым, без ограничений для множества целей, как для частного пользования, так и для коммерческого, которое можно скачать в свободном доступе с сайта разработчиков [4]. Эта платформа, объединяющая моделирование, анимацию, симуляции, композитинг и рендеринг в единой среде, стала доступной альтернативой коммерческим пакетам. Blender позволяет создавать профессиональные анимационные проекты даже на персональных компьютерах среднего класса, что особенно ценно для студий и независимых художников с ограниченным финансированием. Таким образом, компьютерная анимация в России эволюционировала от частного технического приёма до комплексной цифровой среды, объединяющей художественное видение, инженерные решения и производственные стратегии. Переход к использованию игровых движков, открытых платформ и гибридных методов анимации свидетельствует о формировании национальной анимационной инфраструктуры, способной конкурировать на мировом уровне.

3. *Видеомонтаж в эпоху цифровой реальности.* Видеомонтаж в наши дни, принципиально отличается от традиционного кинематографа по структуре творческого процесса. Если в классическом кино художественный замысел в основном реализуется ещё на площадке, то в видеоискусстве ключевой этап формирования образа приходится на финальную обработку. Именно на этом этапе происходит не просто техническая доработка, а рождение новой экранной реальности, которая может сочетать живые кадры, 3D-графику, анимацию в единой композиции [6]. Переходу к нелинейному цифровому монтажу, предоставляет автору беспрецедентную свободу в работе с изображением. В отличие от аналоговых технологий, где монтаж был механически привязан к последовательности записи, цифровое пространство позволяет свободно манипулировать временем, цветом и текстурой. В результате возникает то, что Гук А.А. определяет, как «визуальный гибрид» – синтетическое изображение, одновременно фотореалистичное и фантазийное [6]. Так же с концептуальными и эстетическими трансформациями, видеомонтаж опирается на мощную технологическую базу, ключевым элементом которой остаётся Adobe Premiere Pro – видео редактор для профессионального нелинейного монтажа. Программа давно рекомендовала себя в индустрии: её используют такие медиа компании, как BBC и The Tonight Show, а в числе фильмов, смонтированных в Premiere Pro, – «Социальная сеть», «Дэдпул», «Через пыль к победе» и «Капитан

Абу Раед». Сегодня программа поддерживает работу с видео в разрешении 4K и выше, обработку 32-битных цветовых пространств (RGB и YUV), редактирование аудиосэмплов на уровне отдельных фреймов, интеграцию VST-плагинов и работу с многоканальным звуком, включая объёмный формат 5.1 surround. Архитектура Premiere Pro обеспечивает совместимость с сотнями видео- и аудиокодексов, что делает его универсальным инструментом в условиях фрагментированного медиа ландшафта. Научное сообщество также отмечает растущую роль этого ПО в профессиональном постпродакшене, что подтверждается исследованиями в области медиа образования [9]. В то же время в индустриальном видеопроизводстве монтаж адаптируется под новые форматы потребления. Согласно аналитическому обзору платформы Kurshub, более 75% мобильного видео трафика приходится на вертикальный контент, а оптимальная длительность короткого видео – от 15 до 45 секунд [8]. Это требует от монтажёров пересмотра композиционных принципов: акцент смещается с панорам на крупные планы, а нарратив строится по принципу «микро-драматургии»

4. *Искусственный интеллект: расширение традиционных подходов.* В последние годы наблюдается интеграция машинного обучения в анимационный рабочий процесс. Искусство создается в цифровом пространстве, видеомонтаж происходит в реальном времени, анимация использует огромные массивы данных. Алгоритмы нейросети применяются для автоматического заполнения промежутков между заданными точками ключевых кадров, очистки ручной анимации от артефактов и генерации фона на основе раскадровок [1].

Применение нейросетей становится неотъемлемой частью в искусстве и дизайне. Например, работы художника Рефика Анадол, который применяет искусственный интеллект как дополнительный инструмент в создании цифрового искусства. Важный аспект новых форм искусства – трансформацию огромных массивов данных в визуальные и звуковые композиции. Например, в проекте «WDCH Dreams» 45 терабайт архивных данных Лос-Анджелесского филармонического оркестра были преобразованы с помощью алгоритмов нейросети в динамичные цифровые образы [10]. Цифровая коллекция Biom Lumina, создана студией Рефик Анадол. Которая добавлена в энциклопедию художника представляет собой живые картины на блокчейне, свечение и звук которых зависит от данных окружающей среды, почвы, атмосферы. Цель возвращение к природе, где природа и код находится в гармонии [12]. Применение в студии Артемия Лебедева в 2018 году нейросети вызвало бурную

реакцию. Дело в том, что ИИ разрабатывал коммерческие проекты. Заказчики не подозревали что работу выполняет далеко не человек. Под псевдонимом Николай Иронов с внушительной базой портфолио получал заказы. В студии акцентировали, что эксперимент говорит о начале масштабной коммерческой автоматизации в мире [11].

В последние месяцы новости о применении искусственного интеллекта и его внедрения все чаще были в новостных источниках. Многие компании стремились заключить контракты и идти в ногу с будущим. Анимационная студия Дисней заключила соглашение с Open AI в ноябре этого года. Что говорит о применении нейросетевых технологий в компании, внедрение GPT чата среди сотрудников на официальном уровне. Кроме того, компании подтвердили общую договоренность по контролю за создание нежелательного контента и соблюдения прав владельцев [9]. В октябре Open AI предоставила поддержку приложений внутри chat GPT. Одним из первых были Photoshop и Figma. AI-плагины в Figma такие как Fig pilot который дает возможность работать с чатом GPT, а с плагином Icons Background Remover можно удалить фон сразу из нескольких изображений одновременно, вдобавок появились новые инструменты: edit with prompt изменяет элементы на изображении по текстовому запросу, а exant image можно расширить фон под нужные пропорции не меняя композицию [14]. В программе Photoshop добавили Neural Filters, с его применением возможна автоматическое удаление объектов, сглаживание кожи, удаление артефактов корректировка цвета. А с Infinite Texture Panel можно выбрать текстуру из библиотеки, в которой более 70000 [15]. Вопрос состоит в том, что, если нейросети генерирует значительную часть визуального контента то, кто является автором? Обеспечение безопасности информационной сферы и цифрового наследия, проблема для традиционных практик, которую нужно решать на государственных уровнях. Искусственный интеллект не заменит художника, поскольку нейросеть обучена алгоритмами и моделями уже существующих данных. Таким образом на данный момент, ИИ выступает кистью для творчества, а человек художником [1].

Заключение

В совокупности, рассмотренные направления свидетельствуют о формировании новой художественной парадигмы, в которой мультимедийные технологии становятся не просто инструментами, а средой, языком и даже философией. Искусство перестаёт быть статичным объектом и превращается в процесс, в котором участвуют технологии, данные, зритель и алгоритмы.

Список литературы

1. Канарейко Д. А. Трансформация культуры в цифровой среде // Эргодизайн. 2022. № 2 (16). URL: <https://cyberleninka.ru/article/n/transformatiya-kultury-v-tsifrovoy-srede> (дата обращения: 17.12.2025).
2. Кривуля Н. Г. Проблемы и стратегии копродукции российских и западных анимационных компаний // Театр. Живопись. Кино. Музыка. 2019. № 3. URL: <https://cyberleninka.ru/article/n/problemy-i-strategii-koproduktsii-rossiyskih-i-zapadnyh-animatsionnyh-kompaniy> (дата обращения: 17.12.2025).
3. Яровая Н. Ю. Анимация в цифровой среде // Вестник ВГИК. 2024. № 1 (59). URL: <https://cyberleninka.ru/article/n/animatsiya-v-tsifrovoy-srede> (дата обращения: 15.12.2025).
4. Афанасьев Г. И., Горячкин Б. С., Тоноян С. А. Практикум по экструдированию моделирования Blender: учебно-методическое пособие. М.: Спутник+, 2018. 22 с.
5. Машковцев Б. А. Союзмультфильм: развитие российской мультипликации // Бизнес-кейсы: менеджмент и маркетинг. 2025. Т. 2. № 1. С. 31. URL: <https://casejournal.ru/index.php/cj> (дата обращения: 17.12.2025).
6. Гук А. А. Поэтика видеомонтажа: формирование нового типа экранного изображения // CyberLeninka. URL: https://cyberleninka.ru/article/n/poetika-videomontazha-formirovaniye-novogo-tipa-ekrannogo-izobrazheniya?spm=a2ty_o01.29997173.0.0.7c365171DP0oKn (дата обращения: 17.12.2025).
7. Томсон О. И., Дамницкая А. В. Видеоарт как экспериментальный вид художественных практик современного искусства // Научные труды Санкт-Петербургской академии художеств. 2023. Вып. 69. С. 91–104. URL: https://cyberleninka.ru/article/n/videoart-kak-eksperimentalnyy-vid-hudozhestvennyh-praktik-sovremennogo-iskusstva?spm=a2ty_o01.29997173.0.0.7c365171DP0oKn (дата обращения: 17.12.2025).
8. Видеомонтаж 2025: какие тренды нельзя игнорировать? // Kurshub. 2025. URL: https://kurshub.ru/journal/blog/videomontazh-2025-kakie-trendy-nelzya-ignorirovat?spm=a2ty_o01.29997173.0.0.7c365171DP0oKn (дата обращения: 17.12.2025).
9. Основы применения видеомонтажа для мультимедийной индустрии / Б. Ч. Ли, О. И. Белозеров // Современные тенденции и проекты развития информационных систем и технологий. Материалы Городской научно-исследовательской конференции. Хабаровск, 2023. С. 84-89. URL: <https://elibrary.ru/item.asp?edn=gqncoj> (дата обращения: 17.12.2025).
10. Исследование Новых Форм в медиаискусстве с использованием больших данных, алгоритмов и искусственного интеллекта. URL: https://www.elibrary.ru/download/elibrary_80512461_23935830.pdf (дата обращения: 17.12.2025).
11. Дружинина О. Б. Цифровые технологии в дизайне: прошлое, настоящее, будущее // Архитектура и дизайн в цифровую эпоху. Москва: МГХПА им. С. Г. Строганова, МАРХИ, ПАХ, 2021. 34 с. URL: https://www.academia.edu/68552391/ARCHITECTURE_AND_DESIGN_IN_THE_DIGITAL_AGE_Collective_monograph_on_the_basis_of_the_materials_of_the_International_scientific_conference (дата обращения: 17.12.2025).
12. URL: <https://dataland.art/collections/biome-lumina> (дата обращения: 17.12.2025).
13. URL: <https://openai.com/index/disney-sora-agreement/> (дата обращения: 17.12.2025).
14. URL: <https://www.figma.com/community/plugin/1309912337130106358/figpilot-ai-chatgpt-and-dalle-3-in-figma> (дата обращения: 17.12.2025).
15. URL: <https://www.adobe.com/products/photoshop/neural-filter.html> (дата обращения: 17.12.2025).
16. Петров С. Т., Тарасов А. А. Цифровое наследие культуры: проблемы формирования, развития и безопасности // История и архивы. 2014. №11 (133). URL: <https://cyberleninka.ru/article/n/tsifrovoe-nasledie-kultury-problemy-formirovaniya-razvitiya-i-bezopasnosti-1> (дата обращения: 17.12.2025).

**РОЛЬ И МЕСТО TKINTER
В СОВРЕМЕННЫХ РЕАЛИЯХ
СОЗДАНИЯ КЛИЕНТСКОЙ ЧАСТИ
ТЕХНОЛОГИЧЕСКИХ ПРОДУКТОВ**

Гурина В. М.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: skald.beregovoi@yandex.ru*

Научный руководитель: Береговой А.В.

Введение

Tkinter это одна из популярных библиотек Python. Слово «tkinter» происходит от фразы «Tk interface». Tkinter – это объектно-ориентированная надстройка языка Python над Tcl/Tk – где Tcl (Tool Command Language) является скриптовым языком, а Tk – библиотекой графических виджетов для этого языка. Его можно представить как переводчик с Python на язык Tcl. Мы пишем программу на Python, а код модуля Tkinter переводит ваши инструкции на язык Tcl, который понимает библиотека Tk [1].

Tcl здесь играет роль универсального исполнителя команд – это интерпретируемый язык, специально созданный для встраивания в приложения. Он работает как «движок» для Tk, выполняя низкоуровневые операции по созданию и управлению графическими элементами интерфейса. Вся суть кроссплатформенности возможна именно благодаря тому, что Tcl/Tk существует отдельно от Python и уже решал задачи создания GUI на разных операционных системах.

Цель исследования: анализ библиотеки Tkinter и обоснование актуальности её применения в изучении программирования и в профессиональной деятельности для создания приложений и прототипирования интерфейсов.

Материал и методы исследования

В статье использовано сравнение современных фреймворков для создания приложений на языке программирования Python. Было написано веб-приложение, что показывает сочетание эффективности библиотеки и простоты её изучения и применения в реальных проектах.

**Результаты исследования
и их обсуждение**

Важнейшей характеристикой Tkinter является его статус как стандартного компонента экосистемы Python. Данный модуль включается во все официальные дистрибутивы языка, что обеспечивает его доступность без необходимости дополнительной установки зависимостей – в том числе потому, что Tcl/Tk обычно предустановлен на большинстве операционных систем. Эта особенность делает Tkinter универсальным решением, гарантирующим работоспособность графических интерфейсов на раз-

личных операционных платформах, включая Windows, Linux и macOS.

Фундаментальным аспектом работы с Tkinter является его событийно-ориентированная модель. Библиотека функционирует на основе главного цикла обработки событий (main loop), который непрерывно отслеживает пользовательские действия (щелчки мыши, нажатия клавиш, перемещения курсора) и системные сообщения. В ответ на эти события вызываются соответствующие обработчики – callback-функции, предоставленные разработчиком.

Синтаксически работа с Tkinter характеризуется минималистичным подходом. Создание базового графического интерфейса требует написания всего нескольких строк кода, что делает библиотеку особенно доступной для начинающих разработчиков. Однако за этой кажущейся простотой скрывается достаточно мощная и гибкая система, позволяющая создавать сложные интерфейсные решения через комбинацию базовых компонентов и кастомных настроек. Следует отметить, что несмотря на свою универсальность, Tkinter демонстрирует определенную архитектурную консервативность. Библиотека сохраняет обратную совместимость с ранними версиями Python, что обеспечивает стабильность существующих приложений, но одновременно ограничивает внедрение современных паттернов проектирования пользовательских интерфейсов.

Tkinter, за своей репутацией простого инструмента для новичков, скрывает ряд интересных и мощных возможностей, которые позволяют выходить далеко за рамки создания стандартных форм с кнопками. Одной из таких является виджет Canvas (холст). Это не просто область для рисования линий, а целая интерактивная среда, на которой можно размещать графические примитивы, изображения, текст и даже другие окна. С его помощью можно создавать не только диаграммы и графики, но и полноценные анимированные модели, простые 2D-игры или специализированные редакторы, где объекты можно перемещать, масштабировать и удалять с помощью мыши. Например, вы можете визуализировать алгоритмы сортировки, где элементы будут переставляться с анимацией, или создать интерфейс для управления моделью умного дома, где устройства представлены в виде иконок на плане [2].

Другая часто упускаемая из виду возможность – это расширенная система событий (bindings). В Tkinter можно привязать обработчик не только к клику мыши, но и к наведению курсора, нажатию конкретных комбинаций клавиш, изменению размера окна и даже к собственным, сгенерированным программистом событиям. Это открывает путь к созданию чрезвычайно отзывчивых интерфейсов. Не стоит забывать и о модуле ttk, который представляет собой современную версию классических виджетов. Он включает

такие сложные компоненты, как TreeView – идеальный инструмент для отображения табличных или иерархических данных, например, списка файлов с атрибутами или структуры базы данных. Виджет Notebook позволяет организовать интерфейс с вкладками, что критически важно для сложных приложений, где нужно группировать функциональность. А Progressbar и Combobox делают интерфейс интуитивно понятным и профессиональным [3]. А так же гибкая система стилей и тем в ttk позволяет уйти от устаревшего внешнего вида «по умолчанию». Вы можете не просто менять цвета и шрифты, а переопределять внешний вид всех элементов управления, создавая уникальный и современный дизайн, который будет соответствовать бренду или просто радовать глаз. В совокупности эти возможности превращают Tkinter из инструмента для создания прототипов в платформу для построения полноценных, функциональных и визуально приятных десктопных приложений, которые идеально подходят для внутренних инструментов, утилит сопровождения или научных визуализаций, где важна быстрая разработка и минимальные зависимости.

Приложение Tkinter обычно имеет иерархическую структуру [4]:

- Создать корневое окно с помощью конструктора Tk().
- Добавить в корневое окно виджеты, такие как кнопки, надписи и поля ввода.
- Упорядочить виджеты с помощью менеджеров компоновки, таких как pack(), grid() или place().
- Привязывать события к виджетам, чтобы управлять взаимодействием с пользователем.

Рассмотрим простой пример Tkinter, который создает окно с надписью (рисунок):

```
import tkinter as tk
def main():
    # Создание главного окна приложения
    root = tk.Tk()
    root.title(«Мое первое Tkinter приложение»)
    root.geometry(«300x100»)

    # Создание текстовой метки
    label = tk.Label(
        root,
        text=»Hello, Tkinter!»,
        font=(«Arial», 14),
        fg=»blue»,
        bg=»lightyellow»)
    label.pack(pady=20)

    # Запуск главного цикла обработки событий
    root.mainloop()

# Запуск главной функции
if __name__ == «__main__»:
    main()
```



Простое приложение с библиотекой Tkinter

Несмотря на то, что Tkinter традиционно позиционируется как инструмент для десктопных приложений, он обладает комплексом возможностей, релевантных для задач фронтенд-разработки в широком понимании этого термина [5]:

1) Tkinter имплементирует компонентный подход, где каждый элемент интерфейса (виджет) представляет собой самостоятельную сущность с инкапсулированным состоянием и поведением. Это напрямую коррелирует с компонентной моделью React или Vue.js. Например, виджет Button в Tkinter:

- Обладает свойствами (text, state, command)
 - Имеет собственный жизненный цикл (создание, конфигурация, уничтожение)
 - Поддерживает систему событий (bindings)
- 2) Tkinter предоставляет механизмы для управления состоянием, аналогичные state management в веб-фреймворках:

- Переменные Tkinter (StringVar, IntVar, BooleanVar) как реактивные источники данных
- Автоматическое обновление интерфейса при изменении переменных
- Возможность создания унифицированного хранилища состояния

3) Модуль ttk предлагает систему стилей, концептуально близкую к CSS:

```
style = ttk.Style()
style.configure('Custom.TButton',
    padding=[10, 5],
    font=('Arial', 12),
    background='#3498db')
```

Данный подход позволяет реализовать:

- Централизованное управление внешним видом
 - Каскадное применение стилей
 - Поддержку различных тем оформления
- 4) Tkinter поддерживает принцип композиции через:
- Создание кастомных виджетов на основе существующих
 - Наследование и расширение функциональности
 - Модульную организацию интерфейсных компонентов
- 5) Tkinter естественным образом поддерживает паттерны, характерные для фронтенд-разработки:
- Model-View-Controller через разделение данных и представления

- Observer для реакции на изменения состояния
- Factory для создания сложных компонентов

Пример для Model-View-Controller

```
import tkinter as tk
from tkinter import ttk

class UserModel:
    def __init__(self):
        self.users, self.obs = [], []
    def add_user(self, u, e):
        self.users.append({'username': u, 'email': e})
        for o in self.obs: o.on_change()
    def add_obs(self, o): self.obs.append(o)

class UserView(ttk.Frame):
    def __init__(self, parent, ctrl):
        super().__init__(parent)
        ttk.Entry(self).pack(pady=2); self.entry1 = ttk.Entry(self); self.entry1.pack(pady=2)
        ttk.Entry(self).pack(pady=2); self.entry2 = ttk.Entry(self); self.entry2.pack(pady=2)
        ttk.Button(self, text='Add', command=lambda: ctrl.add_user(self.entry1.get(), self.entry2.get())).pack()
        self.listbox = tk.Listbox(self); self.listbox.pack(fill=tk.BOTH, expand=True)
    def update_list(self, users):
        self.listbox.delete(0, tk.END)
        for u in users: self.listbox.insert(tk.END, f'{u["username"]} - {u["email"]}')

class UserController:
    def __init__(self): self.model, self.view = UserModel(), None
    def set_view(self, v): self.view = v; self.model.add_obs(self)
    def add_user(self, u, e): u and e and self.model.add_user(u, e)
    def on_change(self): self.view and self.view.update_list(self.model.users)

class Settings:
    def __init__(self):
        self._theme, self._lang, self.obs = 'light', 'en', []
    def add_obs(self, o): self.obs.append(o)
    def notify(self, **k): [o.update(**k) for o in self.obs]
    @property
    def theme(self): return self._theme
    @theme.setter
    def theme(self, v): self._theme = v; self.notify(theme=v)
    @property
    def language(self): return self._lang
    @language.setter
    def language(self, v): self._lang = v; self.notify(language=v)

def run_mvc():
    root = tk.Tk(); root.title('MVC')
    c = UserController(); v = UserView(root, c); c.set_view(v)
    v.pack(fill=tk.BOTH, expand=True); root.mainloop()

def run_observer():
    root = tk.Tk(); root.title('Observer'); root.geometry('300x200')
    s = Settings()
    f = tk.Frame(root, height=150); f.pack(fill=tk.BOTH, expand=True)
    lbl = tk.Label(root, text='Theme: light, Lang: en', relief=tk.SUNKEN)
    lbl.pack(fill=tk.X, side=tk.BOTTOM)
    s.add_obs(type('', (), {'update': lambda **k: f.config(bg='black' if k.get('theme') == 'dark' else 'white')})())
    s.add_obs(type('', (), {'update': lambda **k: lbl.config(text=f'Theme: {k.get('theme', 'light')}, Lang: {k.get('language', 'en')}')})())
    for t, cmd in [(('Light', 'light'), (('Dark', 'dark'), ('EN', 'en'), ('RU', 'ru'))):
        attr = 'theme' if t in ['Light', 'Dark'] else 'language'
        tk.Button(root, text=t, command=lambda a=attr, v=cmd: setattr(s, a, v)).pack(side=tk.LEFT)
    root.mainloop()

def main():
    root = tk.Tk(); root.title('Patterns'); root.geometry('250x100')
    for t, cmd in [(('MVC', run_mvc), ('Observer', run_observer), ('Exit', root.destroy))]:
        tk.Button(root, text=t, command=cmd, width=15).pack(pady=2)
    root.mainloop()

if __name__ == '__main__': main()
```

Заключение

На основе проведенного анализа можно констатировать, что Tkinter занимает уникальную и устойчивую нишу в современной экосистеме инструментов для разработки клиентской части технологических продуктов. Несмотря на появление множества альтернативных фреймворков, эта библиотека сохраняет свою практическую значимость благодаря сбалансированному сочетанию простоты, надежности и функциональности. Её ключевые преимущества, такие как нулевой порог вхождения и быстрое освоение, делают Tkinter идеальным решением для образовательных целей и быстрого прототипирования. Минимальные требования к инфраструктуре и встроенная доступность в стандартной поставке Python устраняют сложности с развертыванием и обеспечением совместимости, что критически важно для внутренних инструментов и утилит сопровождения.

Кроссплатформенность библиотеки обеспечивает единообразие поведения приложений в различных операционных средах, что значительно сокращает затраты на тестирование и поддержку. При этом Tkinter демонстрирует достаточную гибкость для реализации сложных архитектурных паттернов, включая MVC, Observer и Factory, что позволяет создавать хорошо структурированные и поддерживаемые приложения. Однако важно отметить, что выбор Tkinter должен быть осознанным и соответствовать конкретным задачам проекта. Для публичных коммерческих продуктов с высокими требованиями к современному пользовательскому опыту более оправданным может быть использование других решений. Но в сценариях, где приоритетом являются скорость разработки, минимальные зависимости и стабильность работы, Tkinter продолжает оставаться оптимальным выбором. Таким образом, библиотека не только сохраняет свою актуальность, но и представляет собой ценный инструмент в арсенале разработчика, особенно в специфических предметных областях, где глубоко интегрирована Python-экосистема.

Список литературы

1. Есилевский С. Знакомство с TCL/TK // Системный администратор. 2021. URL: <https://elibrary.ru/item.asp?id=46486849> (дата обращения: 15.12.2025).
2. Нехорошева Е. М., Конечная Е. А. Графические возможности Ruby // Информационно-телекоммуникационные системы и технологии: материалы Всероссийской научно-практической конференции. Кемерово: Кузбас. гос. техн. ун-т им. Т. Ф. Горбачева, 2021. С. 99–101. URL: <https://elibrary.ru/item.asp?id=49602973> (дата обращения: 15.12.2025).
3. Использование виджета Treeview в Tkinter, вложенные элементы // PythonRu. 2021. URL: <https://pythonru.com/uroki/vidzhet-treeview-tkinter-22> (дата обращения: 15.12.2025).
4. Python Tkinter Tutorial // GeeksforGeeks. 2017. URL: <https://www.geeksforgeeks.org/python/python-gui-tkinter/> (дата обращения: 15.12.2025).
5. Tkinter. Программирование GUI на Python / Светлана Шапошникова (plustilino). 2021.

ГИБРИДНАЯ АРХИТЕКТУРА AI-АССИСТЕНТА ДЛЯ УПРАВЛЕНИЯ ФИНАНСАМИ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ И ПОВЕДЕНЧЕСКОЙ ЭКОНОМИКИ

Денисов Е. Д.

ФГБОУ ВО «Российский экономический университет им. Г. В. Плеханова»,
Москва, e-mail: jenick47@gmail.com

Научный руководитель: Микрюков А. А.

Введение

Современные приложения для учета финансов (например CoinKeeper, Дзен-мани) предназначены для решения задач визуализации динамики, но при этом не обеспечиваются коррекция глубинных моделей поведения. Более 70% решений о покупках являются импульсными или иррациональными, обусловленными когнитивными искажениями, такими как предпочтение сиюминутной выгоды долгосрочной цели, либо покупка «актива» под влиянием новостного фона, а не самостоятельного анализа. Идея предлагаемого подхода заключается в создании цифрового коуча для принятия рационального решения субъектом за счёт синтеза метода машинного обучения (для анализа паттернов) и модели поведенческой экономики для их коррекции. В результате сформирована концептуальная модель алгоритма.

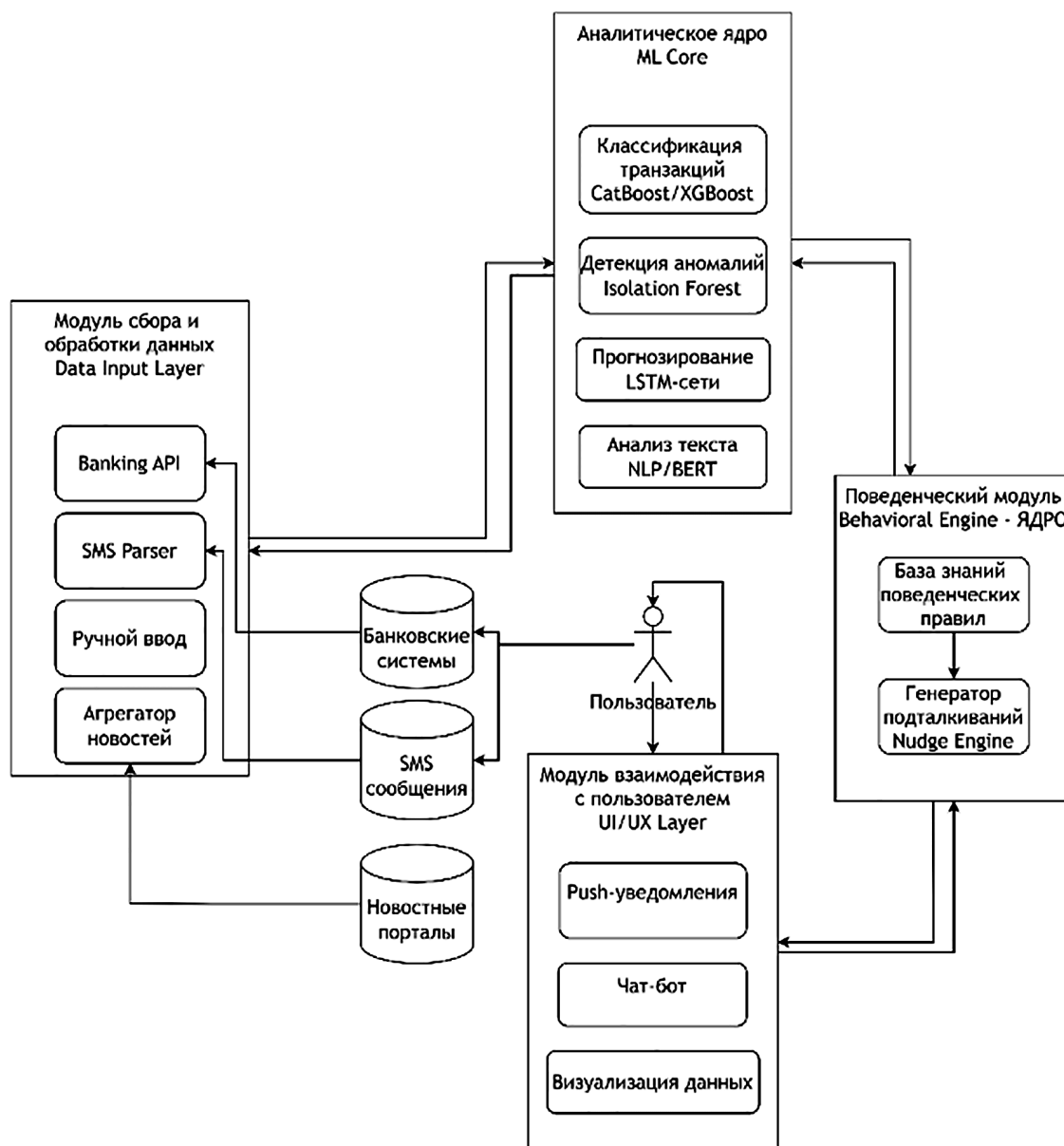
Теоретический анализ рынка

Проведенный анализ существующих решений показывает их принципиальное ограничение, они функционируют как пассивные регистраторы финансовых операций, не оказывая активного воздействия на поведенческие паттерны пользователя. Такие системы предоставляют историческую статистику, но не предотвращают повторение финансовых ошибок.

Возникает противоречие между объективной потребностью пользователей в повышении финансовой дисциплины и неспособностью традиционных приложений преодолеть психологические барьеры, препятствующие рациональному финансовому поведению. Данное противоречие может быть разрешено за счет создания модели, которая фиксирует финансовые операции и активно корректирует поведение пользователя на основе принципов поведенческой экономики.

Концептуальная модель

В ходе исследования была разработана концептуальная модель AI ассистента на основе алгоритма поддержки принятия решений, что реализовано в виде мобильного приложения с серверной частью и состоит из 4 основных модулей (рисунок).



UML-диаграмма компонентов системы

Предлагаемый AI-ассистент представляет собой гибридную модель, где все модули работают в едином контуре управления финансовым поведением.

Модуль сбора данных в реальном времени агрегирует транзакции через банковские API, SMS-парсинг и ручной ввод, формируя полную финансовую картину пользователя. Аналитическое ядро на основе алгоритмов CatBoost и LSTM-сетей автоматически классифицирует транзакции, выявляет аномальные траты и строит прогнозы денежного потока.

Ключевым элементом системы является поведенческий модуль, который анализирует данные через призму когнитивных искажений

(гиперболическое дисконтирование, стадное поведение) и генерирует персонализированные интервенции.

Модуль взаимодействия реализует интервенции через push-уведомления, чат-бота и целеориентированную визуализацию, создавая замкнутый цикл «анализ-решение-коррекция». Например, при обнаружении импульсной покупки, угрожающей долгосрочной цели, система мгновенно проинформирует пользователя на сколько дней отдалится достижение цели и предложит компенсирующие действия.

Рассмотрим кейс пользователя.

Цель – накопить 50 000₽ на новый ноутбук через 4 месяца.

Сценарий:

1. Пользователь заходит в кофейню и покупает кофе за 300Р (ежедневная привычка).

2. Data Layer фиксирует транзакцию через API банка.

3. ML Core классифицирует ее как «Кафе» и помечает как аномалию, но пересчитывает прогноз. Выясняется, что при текущих темпах трат цель по ноутбуку будет достигнута на 2 недели позже.

4. Behavioral Engine сравнивает прогнозируемый остаток средств со стоимостью ноутбука и обнаруживая угрозу достижению цели активирует персонализированное push-уведомление.

5. UI Layer отправляет push-уведомление: «Пользователь, ваша цель «Новый ноутбук» отдалась на 3 дня из-за сегодняшних трат в кафе. Если пропустите кофе еще 2 раза на этой неделе – вернетесь к плану!»

Экономическое обоснование

Внедрение системы потребует первоначальных инвестиций в размере 20 млн рублей на разработку MVP и запуск, а также ежегодных операционных затрат в размере 9 млн рублей на поддержку, облачную инфраструктуру и развитие.

Монетизация осуществляется через freemium-модель с премиум-подпиской за 417 руб./мес. При достижении 10000 активных пользователей с конверсией в платящих 70% годовая выручка составляет:

$7000 \text{ пользователей} \times 5000 \text{ руб. / год} = 35 \text{ млн руб.}$

При чистой прибыли в 14 млн рублей рентабельность достигнет 40%.

$20 \text{ млн руб. / 14 млн руб.} = 1,4 \text{ года}$

Ключевые метрики подтверждают эффективность модели. LTV пользователя 12 000 руб. при SAC 800 руб. обеспечивают устойчивый рост.

Целевой эффект отображает увеличения сбережений на 25-30%.

Экономический эффект для пользователей проявляется в снижении импульсных затрат на 15-20%.

Заключение

Разработана гибридная модель, интегрирующая методы как машинного обучения, так и методы поведенческой экономики, что позволяет достичь целевого эффекта в виде повышения эффективности управления личными финансами. Представлена первая попытка реализации алгоритма поддержки принятия решений, что показала практическую возможность применения данного подхода. Созданная система обеспечивает активную коррекцию финансового поведения пользователей, демонстрируя снижение импульсных затрат на 15-20% и увеличение сбережений на 25-30%. Экономические

расчеты подтверждают рентабельность системы на уровне 40% при сроке окупаемости 1,4 года, что свидетельствует о перспективах дальнейшего развития и возможности коммерциализации решения.

Список литературы

1. Гражданский кодекс Российской Федерации (часть четвертая) от 18.12.2006 № 230-ФЗ (ред. от 31.12.2024) «О персональных данных».

2. Федеральный закон «О национальной платежной системе» от 27.06.2011 № 161-ФЗ (ред. от 02.07.2021).

3. Талер Р., Санстейн К. Nudge. Архитектура выбора. Как улучшать наши решения о здоровье, благосостоянии и счастье. М.: Манн, Иванов и Фербер, 2022. 256 с.

4. Канеман Д. Думай медленно... решай быстро. М.: АСТ, 2023. 624 с.

5. Рассел С., Норвиг П. Искусственный интеллект: современный подход. М.: Вильямс, 2021. 1408 с.

6. Бостром Н. Искусственный интеллект. Этапы. Угрозы. Стратегии. М.: Манн, Иванов и Фербер, 2023. 496 с.

7. Шоломов Л. А. Цифровая трансформация финансовых услуг: учебное пособие. М.: КНОРУС, 2024. 312 с.

8. Петров К. В. Поведенческая экономика в цифровую эпоху. СПб.: Питер, 2024. 288 с.

9. Сидоров А. Н. Машинное обучение в финансовой аналитике. М.: ИНФРА-М, 2023. 415 с.

10. Официальный сайт Банка России. Раздел «Финансовая грамотность». [Электронный ресурс]. URL: https://cbr.ru/protection_rights/finprosvet/ (дата обращения: 05.10.2025).

11. Финтех в России 2024: Глубокий анализ рынка, тренды и инвестиционные перспективы. [Электронный ресурс]. URL: <https://stocklab.ru/fintech-v-rossii-2024-glubokij-analiz-rynka-trendy-i-investiczionnye-perspektivy/> (дата обращения: 10.10.2025).

12. Поведенческие финансы в условиях цифровой экономики. [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/povedencheskie-finansy-v-usloviyah-tsifrovoy-ekonomiki> (дата обращения: 12.10.2025).

13. Машинное обучение в персональных финансовых помощниках: обзор современных подходов. [Электронный ресурс]. URL: <https://habr.com/ru/companies/sberbank/articles/944394/> (дата обращения: 15.10.2025).

14. Машинное обучение в финтехе. [Электронный ресурс]. URL: <https://journal.ugatu.su/index.php/Proceedings/article/view/16742> (дата обращения: 16.10.2025).

15. Безопасность финансовых приложений на Android: как защитить свои деньги в мобильную эру. [Электронный ресурс]. URL: <https://tabsgame.ru/17932-bezopasnost-finansovyh-prilozheniy-na-android-kak-zaschitit-svoi-dengi-v-mobilnyu-eru.html> (дата обращения: 08.10.2025).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ОБЪЁМА ВЫПУСКА ПРОДУКЦИИ ПРЕДПРИЯТИЯ

Ермоленко С. В., Мозговенко А.А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@amozgoenko.ru

В современных условиях развития бизнеса существует острая необходимость в автоматизации процессов управления персоналом и расчёта заработной платы. Существующие решения часто не учитывают специфику конкретных предприятий и требуют доработки под индивидуальные потребности.

Материалы и методы исследования

Современные исследования в области автоматизации бизнес-процессов показывают растущую потребность предприятий в комплексных программных решениях. Особое внимание уделяется разработке систем, способных эффективно управлять кадровыми ресурсами и проводить точный расчёт заработной платы с учётом всех необходимых параметров.

Целью данной работы является:

- представление разработанного программного обеспечения для прогнозирования объёма выпуска продукции;
- демонстрация функциональных возможностей программы;
- описание процесса работы с системой.

Результаты исследования и их обсуждение

В ходе работы над статьей было разработано программное обеспечение для прогнозирования объёма выпуска продукции предприятия.

Проведем поэтапный разбор функционирования созданного программного продукта:

Этап 1. Заходим в созданную программу. На нем указаны данные о проекте, его теме и лице, которое выполняло данную работу. Нижняя часть окна содержит кнопку «Начать работу», которая закрывает заглавное окно и переходит в другое окно, описанное в следующем этапе.

Этап 2. После нажатия кнопки в нижней части заглавного окна «Начать работу» программа выводит следующее окно под названием «Данные работников», показанное на рисунке 1. В верхней части окна находится лента меню и на ней расположены две кнопки «Работники» и «Расчет заработной платы», с помощью которых можно осуществить переход на другую страницу программы. На данный момент, как видно на рисунке 1, после заглавного окна программа привела нас на страницу с перечнем работников, их должностью и окладом. Под перечнем работников расположены три кнопки «Добавить работника», «Редактировать работника» и «Удалить работника».

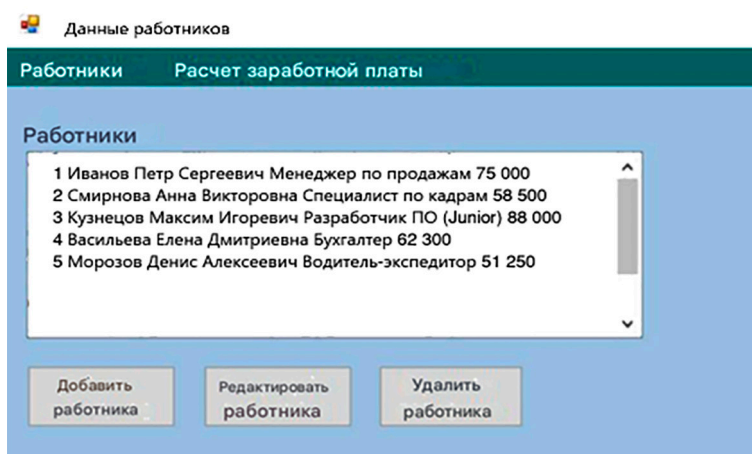


Рис. 1. Данные работников

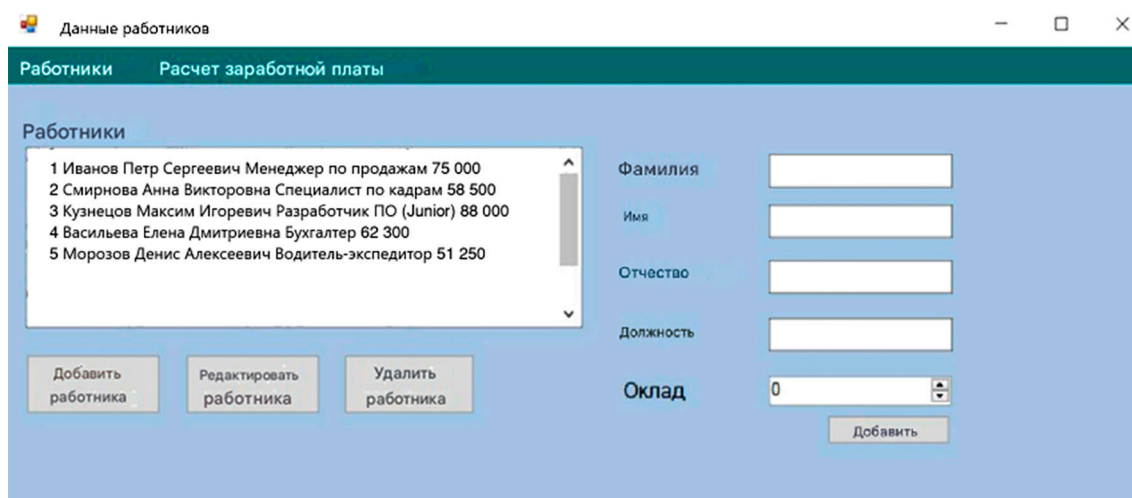


Рис. 2. Добавление нового работника

Этап 3. Далее мы рассмотрим возможность прибавить в существующий список нового работника. Для этого нажимаем кнопку «Добавить работника» и на правой стороне окна у нас появляется пять полей для ввода с названиями «Фамилия», «Имя», «Отчество», «Должность» и «Оклад», а также кнопка «Добавить» (рисунок 2). В представленных полях мы вписываем необходимые данные о новом работнике и, нажав кнопку «Добавить», вносим его в список с другими работниками. При этом следует обратить внимание, что поля Фамилия, Имя, Отчество и Должность обязательно должны быть заполнены, иначе программа не даст добавить работника с неполными данными в список, а названия полей будут подсвечиваться красным цветом.

Если работник уволился, поменял фамилию, у него изменился оклад, он перешел на другую должность или изменились другие данные о нем, разработанный программный продукт позволяет работодателю удалить выбранного работника из списка или отредактировать его данные.

Чтобы отредактировать данные о работнике, необходимо выбрать его из списка, нажав на него левой кнопкой мыши и нажать кнопку «Редактировать работника», после чего в правой части окна появятся поля заполненные существующими данными об этом работнике, которые были введены при добавлении работника в список и кнопка «Редактировать» под. В эти поля, вышедшие с названиями «Фамилия», «Имя», «Отчество», «Должность» и «Оклад», можно ввести необходимые корректировки, изменив любую информацию, указанную в них. После внесения всех необходимых правок нужно нажать кнопку «Редактировать», чтобы сохранить новую информацию о работнике.

Для удаления работника необходимо, так же как и при редактировании его данных, выбрать его в списке, нажав на него левой кнопкой мыши и нажать кнопку «Удалить работника», после этих действий появится окно «Удаление работника», в котором программа предоставляет возможность подтвердить факт удаления, нажав кнопку «Да» или «Нет».

Этап 4. Разобравшись с добавлением, редактированием и удалением работников, перейдем непосредственно к расчетам заработной платы, для чего был создан программный продукт. Для этого перейдем на страницу расчета, нажав кнопку «Расчет заработной платы» в верхней части окна в ленте меню.

В полученном окне (рисунок 3) мы видим в левой стороне окна выпадающие списки с возможностью выбора работника, для которого будут производиться расчеты и возможностью выбора периода, за который будут считаться и уплачиваться заработная плата и налоги за выбранного работника. Другую часть окна занимают поля ввода данных, которые будут использоваться в расчетах. В верхнем ряду отображается должность и оклад выбранного работника и поля для ввода чисел с названиями: «Премия, руб.», «Индексация, %», «Индексация, руб.» в которые вводятся данные. Ниже отображается второй ряд полей для данных, имеющих названия: «По графику», «Отработано», «Больничные», «Отпускные» и «Стаж». В этих полях вводятся данные о количестве дней и, при необходимости, количестве лет стажа работника. Спускаясь ниже, мы видим поля для ввода чисел, которые будут отражать проценты налогов «НДФЛ, %», «ЕСВ, %». Рядом с ними находится кнопка «Рассчитать», которая производит весь расчет на основе введенных в другие поля данных.

Рис. 3. Страница «Расчет заработной платы»

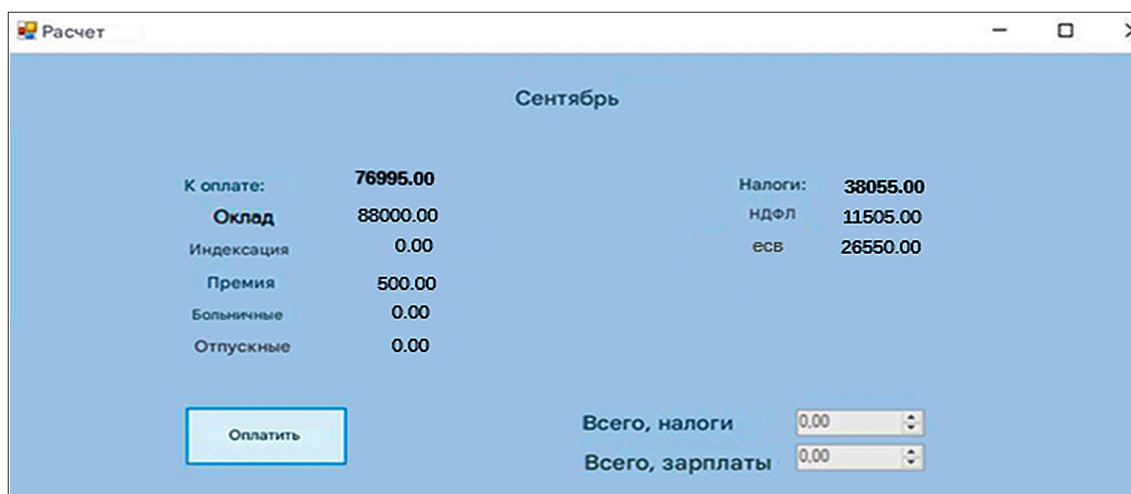


Рис. 4. Итог расчета

Этап 5. Осмотрев программный продукт, попытаемся произвести расчет заработной платы и налогов для любого работника.

Чтобы начать расчет, необходимо выбрать работника, заработную плату которого мы будем считать и период, за который мы должны ему начислить и уплатить налоги. Откроем выпадающий список «Работник», который находится в левой части окна и выберем любого работника, пусть это будет Кузнецов Максим Игоревич, сразу подтянулась информация о его должности (Разработчик ПО) и оклад (88 000 рублей). Затем открываем второй выпадающий список, который находится под предыдущим и называется «Период» и выбираем месяц, за который будут проводиться расчеты, пусть это будет сентябрь 2025 года.

После выбора работника и периода, за который будем производить расчет и выплату заработной платы и налогов, переходим к вводу данных в другие поля.

Предположим, что работнику за этот месяц была начислена премия в размере 500 рублей, поэтому в поле «Премия, руб.» вводим число 500. Индексации у нас нет, поэтому поля «Индексация, %» и «Индексация, руб.» оставляем без изменений. По графику работы работник должен был отработать 20 дней и все дни он присутствовал на работе, поэтому в поля «По графику» и «Отработано» вводим число 20. Поскольку в текущем месяце работник не брал отпуска и начисления пособия по временной нетрудоспособности (больничных) также не нуждался, то в полях «Больничные» значение 0. После этого вводим значение процентов налогов, которые будут удержаны и начислены на ФОТ в поля: «НДФЛ, %» – 13,00, «ЕСВ, %» – 30,00.

После ввода всех необходимых данных для получения результата расчета нажимаем кнопку «Рассчитать» и получаем окно Расчет с результатами вычислений (рисунок 4).

В верхней части окна указан период, за который производился расчет, а ниже представлены два столбца с результатами вычислений. В левом столбце отражены суммы средств за отпуск (0,00 рублей), больничные (0,00 рублей), премия (500,00 рублей), индексация (0,00 рублей), оклад (88 000,00 рублей) и всего к выплате (76995,00 рублей). В правом столбце находятся суммы начисленных налогов: НДФЛ (11505,00 рублей), ЕСВ (26 550,00 рублей) и общая сумма (38 055,00 рублей).

В нижней левой части окна находится кнопка «Оплатить», с помощью которой можно произвести выплату заработной платы работнику и оплатить государственные налоги. В правой нижней части окна отображаются данные обо всех уже уплаченных налогах государству и выплаченной заработной плате работникам предприятия за все периоды вместе.

После получения результата расчетов и оплаты государственных налогов и заработной платы работникам можем закрыть окно «Расчет», нажав на крестик в верхнем правом углу окна. После этого у нас останется открытая страница «Расчет заработной платы» в окне «Данные о работниках». Далее мы можем изменить работника, расчетный период и другие данные для расчета и продолжить исчисление заработной платы и налогов для следующего работника. Также можем полностью выйти из программы, нажав на крестик в верхнем правом углу окна «Данные о работниках».

Все данные по расчетам будут автоматически сохранены.

Выводы

Разработанное программное обеспечение позволяет:

– эффективно управлять данными о сотрудниках предприятия;

- проводить точный расчёт заработной платы с учётом всех необходимых параметров;
- автоматизировать процесс расчёта налогов и других отчислений;
- сохранять и анализировать данные о расчётах.

Программа представляет собой комплексное решение, способное значительно упростить и ускорить процессы управления персоналом и расчёта заработной платы на предприятии. Дальнейшее развитие программного продукта может включать расширение функциональности и адаптацию под специфические требования различных отраслей промышленности.

Список литературы

1. Яковлев А. В. Управление производством: планирование и диспетчеризация. М.: ООО «1С-Пабблишинг», 2018. 219 с. ISBN 978-5-9677-2707-8
2. Разработка информационного обеспечения расчета плановых калькуляций выпускаемой продукции (на примере АО «ККЖБМИ») / Н. Ф. Телешева, И. В. Филимонок, Е. И. Высотенко, Д. И. Ярещенко. Красноярск, 2018. 110 с.
3. Курдин А. Р., Байков М. Ю., Чеблаков Г. С., Пахомова И. В. Влияние глобальных вызовов на российский рынок нефти и нефтепродуктов // Научный журнал НИУ ИТМО. Серия: Экономика и экологический менеджмент. 2020. №1. С. 158-169.
4. Парушина Н. В., Лытнева Н. А., Ершова И. Г. Региональное управление экономикой: монография. Воронеж: Научная книга, 2010. 210 с.
5. Лытнева Н. А. Управление системными изменениями // Вестник ОрелГИЭТ. 2008. №4. С. 72-83.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИНХРОННЫХ И АСИНХРОННЫХ ВЕБ-СЕРВЕРОВ

Журавлёв Д. В., Букреев Д. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru

Научный руководитель: Букреев Д. А.

Введение

В современное время развитие веб-технологий сопровождается ростом требований к производительности серверной части, устойчивости к высокому трафику и способности обрабатывать большое количество одновременных параллельных соединений. Традиционные веб-серверы используют синхронную модель запросов, при которой каждый запрос выполняется последовательно или в отдельных потоках. Такая модель является предсказуемой, при высоком числе одновременных соединений возникает падение производительности.

Современные же веб-приложения выставляют новые требования – поддержку WebSocket-соединений, асинхронных операций и обработку большого числа запросов. Эти условия способствуют развитию событийно-ориентированным, неблокируемым серверам, кото-

рые работают на базе, асинхронных моделей ввода-вывода.

В экосистеме Python синхронная модель реализуется WSGI, среди которых наиболее распространённой является сервер Gunicorn, который демонстрирует высокую стабильность и надёжность при использовании многопроцессной модели обработки запросов, что делает его стандартным решением для классических веб-приложений.

Альтернативой синхронной архитектуре стал асинхронный стек ASGI, который обеспечивает неблокирующую обработку соединений. Один из наиболее изученных представителей является Uvicorn, показывая, что ASGI-серверы демонстрируют существенный прорыв в скорости обработки большого числа параллельных запросов благодаря событийной модели и отсутствию блокировок. Дополнительным элементом серверной архитектуры является Nginx – высокопроизводительный обратный прокси-сервер. Благодаря его модели работы Nginx способен эффективно обслуживать десятки тысяч одновременных соединений с минимальными задержками. Таким образом задача эпохи высоконагруженных веб-систем заключается в сравнительном анализе синхронных и асинхронных моделей обработки запросов, оценке их производительности и архитектуры, выбор оптимального решения для конкретного класса приложений.

Цель исследования – провести систематический сравнительный анализ синхронных и асинхронных веб-серверов, выявить их архитектурные особенности, показатели производительности и области применения, при проектировании современных веб-систем.

Материал и методы исследования

Исследование заключается в сравнительном анализе современных синхронных и асинхронных веб-серверов, применяемых в современной разработке веб-приложений.

В качестве материалов исследования рассматривались, три серверных решения, применяющиеся в современной веб-разработке: синхронный сервер Gunicorn, асинхронный сервер Uvicorn и обратный прокси-сервер Nginx. Для анализа были использованы данные о их времени отклика, возможной пропускной способности, их устойчивости к нагрузке и так же особенности архитектуры и модели обработки запросов.

Методы исследования включают в себя аналитическое изучение принципов работы синхронной и асинхронной обработки запросов, а также сравнение серверов по ключевым показателям производительности.

Дополнительно было произведено тестирование под нагрузкой, которое позволило оценить работу серверов в условиях нагрузки реального трафика.

Результаты исследования и их обсуждение

Проведенное исследование было направлено на сравнительный анализ поведения синхронных и асинхронных веб-серверов в условиях различной нагрузки, а также на оценку влияния прокси-уровня Nginx на общую производительность системы. Результаты были получены с опорой на данные экспериментальных и обзорных работ, которые посвящены архитектуре высоконагруженных приложений, особенностям асинхронной обработки и сравнительному анализу веб-серверов [1-3].

1. Поведение синхронных веб-серверов при увеличении нагрузки

В ходе проведенного исследования было установлено, что синхронные веб-серверы, использующие модель блокирующей обработки запросов, характерную для классического WSGI-подхода, демонстрируют стабильные и предсказуемые показатели производительности при сравнительно небольшом количестве одновременных клиентских соединений. В условиях низкой и умеренной нагрузки такие серверы способны обеспечивать приемлемое среднее время отклика, при этом пропускная способность системы возрастает практически линейно по мере увеличения числа активных пользователей вплоть до определенного порогового значения. Однако при дальнейшем росте уровня параллелизма начинают проявляться характерные ограничения блокирующей архитектуры. Среднее время отклика заметно увеличивается, что негативно сказывается на общей производительности системы, а рост пропускной способности замедляется и со временем выходит на плато. Это связано с тем, что при достижении максимального количества доступных рабочих потоков сервер перестает эффективно обрабатывать новые запросы, вынужденно переводя их в очередь ожидания.

Наиболее выраженные недостатки синхронной модели обработки запросов проявляются в сценариях, где наблюдается существенное увеличение времени удержания соединений. Подобные ситуации характерны для веб-приложений, выполняющих длительные операции ввода-вывода, включая обращения к базам данных, взаимодействие с удаленными внешними сервисами, работу с файловыми системами и сетевыми ресурсами. В рамках блокирующей модели каждый такой запрос занимает отдельный рабочий поток на весь период выполнения операции, включая время ожидания завершения ввода-вывода.

В результате сервер оказывается вынужден последовательно обрабатывать запросы в рамках ограниченного пула потоков, что приводит к формированию очередей и росту времени ожидания для клиентов. Данное обстоятельство напрямую отражается на качестве пользовательского опыта, выражаясь в увеличении задержек,

снижении отзывчивости системы и потенциальных тайм-аутах соединений [2].

Таким образом, можно сделать вывод, что синхронные серверные решения, основанные на традиционной архитектурной модели «один поток – один запрос», обладают ограниченными возможностями масштабирования в условиях высокой конкурентности. При увеличении числа одновременных операций такая архитектура приводит к нерациональному использованию вычислительных ресурсов, перерасходу памяти и снижению общей эффективности задействования аппаратных возможностей сервера, что делает ее менее подходящей для современных высоконагруженных веб-приложений.

2. Эффективность асинхронных серверов и событийно-ориентированных архитектур

В процессе тестирования асинхронных серверных конфигураций были получены результаты, демонстрирующие принципиально иное поведение системы при увеличении числа одновременных клиентских соединений по сравнению с синхронными решениями. Асинхронные серверы, использующие событийно-ориентированную архитектуру и неблокирующие операции ввода-вывода, показали способность сохранять низкий уровень задержек на протяжении значительно более широкого диапазона нагрузок. По мере роста числа подключенных клиентов среднее время отклика увеличивалось более плавно и равномерно, не демонстрируя резких скачков, характерных для блокирующих моделей обработки запросов. Пропускная способность асинхронного сервера при этом продолжала возрастать вплоть до уровней нагрузки, при которых синхронные серверы уже испытывали выраженную деградацию производительности, проявляющуюся в росте очередей запросов и увеличении времени ожидания отклика [5]. Данный эффект объясняется тем, что асинхронная модель не требует закрепления отдельного потока выполнения за каждым запросом на весь период его обработки, что позволяет более эффективно использовать вычислительные ресурсы сервера.

На основании полученных результатов, а также анализа особенностей асинхронной обработки запросов в веб-приложениях, можно сделать вывод о том, что переход к неблокирующей архитектуре позволяет существенно сократить простои, связанные с ожиданием завершения операций ввода-вывода. Взаимодействие с внешними ресурсами, такими как базы данных, сетевые сервисы и файловые системы, перестает приводить к блокировке рабочих потоков, что способствует повышению общей утилизации серверных ресурсов без необходимости пропорционального увеличения вычислительной мощности или объема оперативной памяти. Особенно отчетливо преимущества асинхронных серверов проявляются в сценариях с дли-

тельно удерживаемыми соединениями, например при использовании протоколов WebSocket, реализации стриминговых механизмов передачи данных или работе с серверными событиями. В подобных условиях сервер способен приостанавливать обработку конкретного соединения и переключаться на выполнение других задач, возвращаясь к нему только при возникновении соответствующего события, вместо блокировки потока на все время существования соединения. Данный подход обеспечивает более равномерное распределение нагрузки между доступными ресурсами и позволяет обслуживать значительно большее количество клиентов при фиксированной аппаратной конфигурации сервера. Именно поэтому событийно-ориентированная и асинхронная модели обработки запросов лежат в основе большинства современных архитектур высоконагруженных и масштабируемых веб-приложений, ориентированных на работу в условиях высокой конкурентности и интенсивного взаимодействия с внешними ресурсами.

3. Влияние использования Nginx как прокси-уровня

Отдельное внимание в рамках исследования было уделено анализу влияния обратного прокси-сервера Nginx, используемого в качестве внешнего прокси уровня, размещенного перед серверами приложений, на общую производительность и устойчивость системы. Полученные экспериментальные данные подтверждают, что включение Nginx в архитектуру веб-приложения позволяет существенно повысить стабильность функционирования системы и увеличить ее совокупную пропускную способность. Использование Nginx способствует эффективной разгрузке серверов приложений за счет переноса на прокси-уровень обработки статического контента, выполнения терминации TLS-соединений, а также первичной маршрутизации и балансировки входящих HTTP-запросов. Благодаря событийно-ориентированной архитектуре и неблокирующей модели обработки соединений, Nginx способен обслуживать десятки тысяч одновременных подключений при относительно низком потреблении вычислительных ресурсов, что является критически важным фактором для высоконагруженных систем, особенно в условиях кратковременных или непредсказуемых всплесков трафика [4]. В случае использования синхронных серверов приложений внедрение Nginx оказывает выраженный положительный эффект, позволяя сглаживать пиковые нагрузки и сокращать количество прямых соединений, поступающих непосредственно на сервер приложений. За счет этого снижается вероятность исчерпания пула рабочих потоков, уменьшается глубина очередей запросов и повышается устойчивость системы к резким изменениям интенсивности нагрузки. В результате наблюдается снижение среднего времени

отклика и более плавный, контролируемый рост задержек по мере увеличения числа клиентов.

В конфигурациях, основанных на асинхронных серверах приложений, применение Nginx дополнительно усиливает уже присущие им преимущества событийной архитектуры. Комбинация неблокирующего прокси-уровня и асинхронного сервера приложений формирует согласованную многоуровневую архитектуру, в которой каждый компонент эффективно обрабатывает свою часть нагрузки без взаимной блокировки ресурсов. Такая интеграция обеспечивает максимальные показатели пропускной способности и высокую устойчивость к нагрузкам, сохраняя при этом низкий и предсказуемый уровень задержек даже при значительном увеличении числа одновременных соединений. Таким образом, результаты исследования показывают, что использование Nginx в качестве обратного прокси-сервера является важным архитектурным элементом при проектировании современных высокопроизводительных и масштабируемых веб-приложений. Особенно эффективной данная схема оказывается при сочетании Nginx с асинхронными серверами приложений, где достигается оптимальный баланс между производительностью, устойчивостью и рациональным использованием аппаратных ресурсов.

Заключение

В ходе анализа полученных результатов исследования, подтверждается, что синхронные веб-сервера имеют свою практическую значимость в задачах с относительно простой моделью загрузки: небольшое число одновременных пользователей, отсутствие долговременных соединений. В таких условиях блокирующая модель не доходит до пика своих ограничений и может работать быстро и стабильно. Преимущества такой модели проявляются в простоте конфигурации и предсказуемости поведения. Было выявлено, что асинхронные веб-сервера и событийно-ориентированные архитектуры демонстрируют более устойчивое состояние в высоконагруженных системах. При большем количестве одновременных соединений, так же при наличии долгоживущих запросов и при значительной части времени, которая затрачивается на взаимодействие с внешними ресурсами, поэтому переход с блокирующей модели на неблокирующую модель обеспечивает более высокую производительность, пропускную способность и меньшие задержки. Подтверждается, что внедрение Nginx в качестве прокси-уровня, является хорошим и эффективным решением, для повышения производительности и надежности всей системы, не зависимо от типа сервера, который будет использоваться на уровне приложений. Но стоит учесть, что наиболее заметный эффект наблюдается при сочетании Nginx с асинхронными серверами, где общая со-

бытийная модель прослеживается на всем пути обработки запросов.

Список литературы

1. Амиров С. Н. Особенности разработки высоконагруженных систем // International Journal of Open Information Technologies. 2020. №8. URL: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-vysokonagruzhennyh-sistem> (дата обращения: 11.12.2025).
2. Букреев Д. А., Луц С. М. Особенности разработки web-платформ автоматизированного взаимодействия с клиентом // Современные проблемы геометрического моделирования и информационные технологии: материалы II Межрегиональной научно-практической конференции преподавателей и студентов, посвященной 60-летию образования Мелитопольской школы прикладной геометрии (Мелитополь, 28 мая 2024 года). Мелитополь: Мелитопольский государственный университет, 2024. С. 76-85. EDN: BVUZYR.
3. Журавлев Д. В., Букреев Д. А. Современное состояние реактивных технологий frontend разработки // Современные проблемы геометрического моделирования и информационные технологии: материалы II Межрегиональной научно-практической конференции преподавателей и студентов, посвященной 60-летию образования Мелитопольской школы прикладной геометрии (Мелитополь, 28 мая 2024 года). Мелитополь: Мелитопольский государственный университет, 2024. С. 152-157. EDN: EGZAEХ.
4. Латыпов Э. Ф. Сравнительный анализ работы веб-серверов apache и nginx // Экономика и социум. 2017. №6-2 (37). URL: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-raboty-veb-serverov-apache-i-nginx> (дата обращения: 11.12.2025).
5. Хомякова А. А. Возможности программного обеспечения «веб-серверов» Apache и MS IIS по противодействию деструктивным информационным кибернетическим воздействиям // Новые информационные технологии в автоматизированных системах. 2019. № 22.

АНАЛИЗ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ РАЗРАБОТКИ И ПРОЕКТИРОВАНИЯ ВЕБ-РЕСУРСА

Задорожный А. Я., Букреев Д. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru

Научный руководитель: Букреев Д. А.

Введение

Стремительное развитие цифровых технологий привело к существенному усложнению задач, связанных с проектированием и разработкой веб-ресурсов. Если ранее веб-сайт выполнял преимущественно информационную функцию и строился на основе минимального набора технологий, то сегодня он представляет собой полноценную программную систему, обеспечивающую интерактивное взаимодействие, обработку данных, интеграцию со сторонними сервисами и адаптацию под различные пользовательские устройства. Рост пользовательских ожиданий в отношении удобства, скорости загрузки и безопасности существенно влияет на выбор инструментов и технологий, а сама веб-разработка становится междисциплинарным направлением, объединяющим программирование, проектирование интерфейсов, архитектуру приложений и инфраструктурные решения.

Современный веб-ресурс должен одновременно удовлетворять целому ряду требований: обеспечивать понятную навигацию и визуальную целостность интерфейса, корректно отображаться на различных устройствах, быстро реагировать на действия пользователя и гарантировать надёжную защиту данных. Достижение такого сочетания характеристик возможно только при грамотном выборе технологического стека и архитектурных подходов. Анализ современных технологий разработки позволяет выявить ключевые тенденции отрасли, связанные с переходом к одностраничным приложениям, широким распространением реактивных фреймворков, использованием гибких архитектурных подходов и развитием инструментов автоматизации. Значимую роль играет и этап проектирования веб-ресурса, включающий UX/UI-аналитику, создание прототипов, оценку пользовательских сценариев и выбор архитектурного решения. Именно сочетание технологической и проектной составляющих определяет качество конечного продукта.

Цель исследования – анализ современных технологий разработки и проектирования веб-ресурсов, выявление их особенностей, преимуществ и ограничений, а также формирование рекомендаций по выбору технологического стека в зависимости от задач веб-проекта.

Материалы и методы исследования

Материалом исследования послужил комплекс современных технологий, применяемых при проектировании и разработке веб-ресурсов, а также нормативные и методические документы, описывающие требования к веб-приложениям с точки зрения функциональности, удобства использования, производительности и безопасности. В обзор были включены как классические технологии, лежащие в основе веба (HTML5, CSS3, стандарты ECMAScript), так и современные фреймворки фронтенд- и бэкенд-разработки, архитектурные подходы к построению веб-приложений, инструменты автоматизации и средства обеспечения качества. Дополнительно учитывался практический опыт реализации веб-ресурсов образовательного и информационного профиля, что позволило сопоставить теоретические положения с реальными сценариями разработки.

Методологическая основа исследования опирается на сочетание теоретического анализа и сравнительного подхода. Дополнительно использовались элементы системного подхода и структурного анализа: технологии рассматривались не изолированно, а как части единого технологического стека, включающего этапы проектирования, реализации, тестирования и сопровождения веб-ресурса. Это позволило оценить совместимость инструментов между собой, а также определить типичные конфигу-

рации стеков, применяемых на практике. Обобщение полученных результатов осуществлялось посредством аналитического синтеза, в ходе которого были сформулированы выводы и рекомендации по выбору технологий для разработки и проектирования современных веб-ресурсов.

Результаты исследования и их обсуждение

Развитие веб-технологий отражает общую динамику цифровой среды, которая за сравнительно короткий период прошла путь от простых текстовых страниц к сложным интерактивным системам, обеспечивающим полноценную пользовательскую функциональность. Сегодня веб-ресурс уже не рассматривается как статический источник информации – он воспринимается как программный продукт со своей архитектурой, логикой взаимодействия и требованиями к качеству. Поэтому анализ современного состояния веб-технологий предполагает не только исторический обзор, но и понимание тех факторов, которые определяют подходы к проектированию и разработке современных веб-систем.

Первые веб-страницы были строго статичными и решали одну задачу – предоставление информации. HTML в ранних версиях обеспечивал лишь базовую структуру текста, а возможности оформления и взаимодействия были минимальны. На этом этапе именно доступность и простота являлись основными преимуществами веба.

Однако по мере роста аудитории и усложнения пользовательских запросов появилась необходимость в более динамичном содержимом. Появление JavaScript и развитие CSS фактически изменили характер веба: стало возможным не только управлять внешним видом страниц, но и реагировать на действия пользователя, обновлять данные без перезагрузки и создавать первые элементы интерактивности.

Эффективная разработка веб-ресурса начинается задолго до появления первой строки кода. Современная практика подчёркивает значимость проектирования как самостоятельного этапа, от которого напрямую зависит удобство взаимодействия пользователя с системой, её архитектурная устойчивость и дальнейшая масштабируемость. Веб-проектирование объединяет аналитическую, дизайнерскую и инженерную составляющие, что требует от разработчика не только владения инструментами, но и понимания логики построения пользовательских сценариев, особенностей работы интерфейса и архитектурных принципов будущего приложения.

Современный подход к проектированию веб-ресурсов строится вокруг идеи, что успешный продукт – это не просто корректно работающий сайт, а удобный, понятный и технически надёжный инструмент, который воспринимается пользователем естественно и предсказуемо.

Поэтому на ранних этапах особое значение приобретает разработка структуры взаимодействия, а также выбор архитектурного решения, позволяющего обеспечить устойчивую работу приложения независимо от роста нагрузки или расширения функционала [1].

UX/UI-проектирование и прототипирование

Пользовательский опыт (UX) играет ключевую роль в формировании цифровых продуктов. Прежде чем приступить к разработке интерфейса, важно понять, каким способом пользователь будет взаимодействовать с системой, какие действия для него являются приоритетными и какие сценарии должны быть выполнены с минимальными усилиями. На основе таких требований формируются пользовательские пути, определяется логика переходов и выбираются основные паттерны интерфейса.

Инструменты прототипирования – такие как Figma, Adobe XD или Sketch – позволяют визуализировать структуру будущего веб-ресурса ещё до начала разработки. Преимущество такого подхода заключается в возможности быстро проверять идеи, корректировать структуру и обсуждать варианты взаимодействия с командой разработки и заказчиками, не прибегая к затратной переработке кода.

Параллельно создаются UI-макеты, в которых разрабатываются визуальные компоненты – сетка, типографика, цветовая схема, элементы управления. Важным аспектом является соответствие дизайн-системам, которые обеспечивают единообразие интерфейса и упрощают дальнейшую разработку.

Архитектурные подходы к разработке веб-ресурсов

После определения структуры интерфейса возникает вопрос выбора архитектуры, которая обеспечит устойчивость и гибкость будущей системы. Развитие веб-технологий привело к многообразию архитектурных решений, каждое из которых по-своему отвечает вызовам современного веба [2].

Традиционная монолитная архитектура всё ещё применяется в небольших проектах, где важна простота развертывания и целостность кода. Однако с увеличением функциональности она быстро становится громоздкой и менее удобной для сопровождения. На смену монолиту пришли микросервисные подходы, в которых крупная система разделяется на самостоятельные сервисы, взаимодействующие через API. Для веб-приложений это означает возможность независимого обновления модулей, масштабирования отдельных компонентов и гибкого распределения нагрузки.

Ещё один значимый тренд – микрофронтенды, позволяющие разделять клиентскую часть большого приложения между несколькими

ми командами и технологиями. Такой подход особенно актуален для крупных организаций, где разные модули интерфейса развиваются автономно. Отдельного внимания заслуживает архитектурная концепция JAMstack, ориентированная на ускорение загрузки и повышение безопасности за счёт разделения статического рендеринга интерфейса и динамических функций, вынесенных в облачные сервисы. Этот подход активно применяется при создании ресурсных сайтов, маркетинговых платформ и приложений, ориентированных на быструю доставку контента. Каждая архитектура предлагает своё сочетание преимуществ, и выбор зависит от характера проекта, его функциональных требований и перспектив развития.

Для осмысления современной фронтенд-среды важно рассматривать её не как набор отдельных технологий, а как экосистему, где каждый компонент играет свою роль: одни отвечают за структуру и стиль, другие – за реактивность и шаблоны, третьи – за сборку и оптимизацию. Такое взаимодействие формирует основу пользовательского опыта, на который опирается всё приложение. В основе фронтенда по-прежнему лежат три стандарта: HTML, CSS и JavaScript. Однако их современное состояние серьёзно отличается от того, каким оно было даже десять лет назад.

Ключевым шагом в развитии фронтенда стало появление реактивных фреймворков, которые стандартизировали способы создания сложных интерфейсов.

1. React, созданный компанией Meta, предложил компонентный подход, в котором каждый элемент интерфейса рассматривается как самостоятельный модуль со своим состоянием.

2. Vue.js сделал акцент на простоте внедрения и постепенном наращивании функциональности.

3. Angular, напротив, представляет собой комплексный фреймворк, включающий инструменты маршрутизации, работы с формами, модульной организации кода.

Последние годы отмечены ростом интереса к более лёгким решениям, таким как Svelte, который выносит большую часть работы на этап сборки, что делает итоговый код компактнее и быстрее. Фреймворки перестали быть просто инструментами – они стали платформами для создания интерфейсов, задающими архитектурные и организационные решения.

Современная разработка невозможна без инструментов, которые обеспечивают удобство написания и оптимизацию итогового кода. Большинство проектов строится на экосистеме npm и использует сборщики – будь то Webpack, Parcel, Rollup или более современные решения вроде Vite.

Эти инструменты:

– упрощают структуру проекта;

– позволяют использовать модульность кода;

– оптимизируют загрузку ресурсов;

– обеспечивают живую перезагрузку интерфейса при разработке;

– поддерживают интеграцию TypeScript, препроцессоров и тестовых библиотек.

Параллельно развиваются средства анализа кода, тестирования и форматирования: Jest, Cypress, ESLint, Prettier. Они становятся частью культуры разработки, формируя привычку поддерживать качество и читаемость интерфейсной части.

Рост мобильного трафика сделал адаптивность не дополнительной функцией, а обязательным условием. Современные веб-ресурсы проектируются по принципу mobile-first, что означает первичную ориентацию на небольшие экраны. Использование гибких сеток, медиа-запросов, относительных единиц измерения и адаптивных компонентов интерфейса стало нормой. Библиотеки наподобие Bootstrap или Tailwind CSS предлагают готовые решения, которые позволяют ускорить разработку и обеспечить единообразие дизайна.

Адаптивность сегодня – это не только подгонка интерфейса под разные устройства, но и оптимизация загрузки ресурсов, поддержка жестов, снижение энергопотребления и корректная работа в условиях ограниченного канала связи. Если фронтенд определяет то, как пользователь видит и воспринимает веб-ресурс, то бэкенд формирует основу его функциональности. Именно серверная часть отвечает за обработку данных, управление логикой приложения, безопасность, взаимодействие с внешними сервисами и хранение информации. Из-за роста требований к производительности и масштабируемости бэкенд-разработка стала направлением, где активно развиваются новые платформы, архитектурные подходы и инструменты автоматизации.

Современный бэкенд представляет собой сочетание языков программирования, фреймворков, сетевых протоколов, систем хранения данных и инфраструктурных решений. Вместе они образуют технологический фундамент, на котором строится веб-приложение, и именно от качества этих решений зависит стабильность и надёжность системы. Разнообразие серверных платформ объясняется различными требованиями проектов: от быстрых прототипов до высоконагруженных корпоративных систем [7].

1. Node.js стал одним из наиболее распространённых решений благодаря своей событийной модели и использованию одного языка – JavaScript – по всему стеку разработки.

2. Python через фреймворки Django и Flask привлекает разработчиков лаконичностью синтаксиса и обширной экосистемой библиотек.

3. PHP, несмотря на долгую историю критики, остаётся актуальным благодаря устойчивой

инфраструктуре и популярным фреймворкам, таким как Laravel.

4. Java и C# традиционно используются в крупных корпоративных системах, где важна строгая типизация, устойчивость и производительность.

5. Go привлекает вниманием благодаря своей лёгкости, низким накладным расходам и высокой производительности.

Фреймворки определяют структуру серверного приложения и существенно ускоряют разработку:

1. В экосистеме Node.js широко используется Express, который обеспечивает гибкость и минималистичную архитектуру. Более структурированным подходом обладает NestJS, объединяющий идеи модульности и строгой типизации.

2. В Python-среде Django остаётся наиболее комплексным решением: встроенная ORM, панель администратора, средства авторизации и валидации данных позволяют быстро создавать прототипы и полноценные веб-приложения. Flask выбирают в проектах, где важна лёгкость и минимализм.

3. В мире PHP доминирует Laravel, который предлагает удобный синтаксис, развитые инструменты для миграций БД, маршрутизации и шаблонизации. Это делает его одним из наиболее популярных фреймворков для разработки веб-сервисов.

4. Java-разработка традиционно опирается на Spring, представляющий собой универсальную платформу для корпоративных приложений, микросервисов и сложных распределённых систем.

Каждый из этих фреймворков формирует собственный подход к построению архитектуры проекта, и выбор конкретного решения определяется масштабом и требованиями веб-ресурса.

Современные веб-приложения редко ограничиваются статичными страницами, поэтому важной частью бэкенда является организация взаимодействия между клиентом и сервером [6].

Наиболее распространённым подходом остаётся архитектурный стиль REST, который обеспечивает понятный обмен данными с использованием стандартных HTTP-методов. Его преимущества – простота, универсальность и широкий набор инструментов. Альтернативой REST в последние годы стал GraphQL, позволяющий клиенту запрашивать только те данные, которые ему необходимы. Это снижает объём передаваемой информации и делает интерфейсы гибкими, особенно для сложных клиентских приложений. Для задач реального времени используются WebSocket-соединения, обеспечивающие двусторонний канал взаимодействия. Они незаменимы в приложениях, требующих мгновенной реакции – чаты, онлайн-игры, мониторинговые панели. Таким образом, выбор

протокола определяется характером данных, нагрузкой и требованиями к скорости обмена [5].

Работа с данными – ключевой элемент серверной разработки. Выбор системы хранения зависит от структуры данных, масштаба проекта и требований к скорости обработки. Реляционные базы данных, такие как PostgreSQL и MySQL, сохраняют популярность благодаря надёжности, поддержке транзакций и гибкости запросов. Они хорошо подходят для систем, где важна согласованность данных. NoSQL-базы, включая MongoDB [4], Redis и Cassandra, ориентированы на гибкость структур данных, высокую скорость операций и горизонтальное масштабирование [3]. Они применяются в аналитических системах, веб-приложениях с активными пользовательскими данными и сервисах реального времени. Использование ORM-инструментов (например, SQLAlchemy, Prisma, Hibernate) облегчает работу с базами данных, снимая необходимость писать низкоуровневые запросы и позволяя сосредоточиться на логике приложения.

Современная разработка невозможна без инструментов, обеспечивающих поддержку процесса развертывания и сопровождения приложения.

1. Docker стал стандартом контейнеризации: он позволяет запускать приложение в одинаковой среде независимо от рабочего окружения разработчика или сервера. Это повышает устойчивость и предсказуемость развёртывания.

2. Системы CI/CD, такие как GitHub Actions, GitLab CI и Jenkins, автоматизируют тестирование, сборку и публикацию приложения. Это уменьшает человеческий фактор и ускоряет выпуск обновлений.

3. Использование облачных платформ (AWS, Azure, DigitalOcean) позволяет масштабировать веб-ресурс по мере роста нагрузки, обеспечивая распределение трафика, резервирование данных и гибкое управление инфраструктурой.

Практическая разработка веб-ресурса представляет собой последовательность шагов, которые, несмотря на различия в масштабах и специфике проектов, подчиняются общим закономерностям. На этом этапе теоретические решения – выбор архитектуры, стек технологий, проектирование интерфейсов – начинают воплощаться в конкретных технических действиях. Именно здесь становится особенно заметным, насколько удачно были сформулированы требования, насколько рационально выбран технологический стек и насколько продуманной оказалась архитектурная схема. Современная веб-разработка предполагает опору на гибкие процессы, позволяющие адаптировать продукт к изменяющимся условиям. Это требует не только технической компетентности, но и умения организовать разработку таким образом, чтобы каждый этап был логически связан с пре-

дыдущим и одновременно оставался открытым для изменений. Выбор технологий влияет не только на скорость разработки, но и на дальнейшую поддержку проекта. Разработчики ориентируются на несколько факторов: тип создаваемого веб-ресурса, предполагаемую нагрузку, требования к безопасности и взаимодействию с внешними сервисами.

Для информационных порталов или образовательных платформ важна устойчивость и гибкость – здесь нередко используется сочетание React или Vue для интерфейсной части и Django, Laravel или Node.js для серверной логики. В проектах, связанных с потоковыми данными или реалтайм-взаимодействием, предпочтение может отдаваться Node.js или Go. Критерии выбора включают зрелость технологии, активность сообщества, наличие документации, возможность масштабирования и соответствие долгосрочным планам развития продукта. В этом контексте стек перестаёт быть простым набором инструментов – он становится стратегическим решением.

Одним из ключевых элементов практической разработки является тестирование. Оно позволяет не только выявлять ошибки, но и предотвращать появление новых дефектов при дальнейшем развитии проекта. В современной практике используется несколько уровней тестирования:

- модульное, ориентированное на отдельные функции;
- интеграционное, проверяющее взаимодействие частей системы;
- сквозное (end-to-end), моделирующее реальные пользовательские сценарии;
- нагрузочное, оценивающее устойчивость под высокой нагрузкой.

Помимо тестирования, большое значение имеет автоматизация процессов – использование CI/CD, автоматической сборки, проверок качества кода. Это снижает зависимость результатов от человеческого фактора и делает процесс разработки более предсказуемым. Безопасность рассматривается как обязательный элемент современного веб-ресурса. Нарушение в этой сфере приводит не только к техническим сбоям, но и к потере доверия пользователей, что зачастую наносит больший ущерб. Поэтому при разработке учитываются рекомендации OWASP, применяются методы защиты от XSS- и CSRF-атак, реализуются механизмы шифрования данных, а процессы аутентификации и авторизации выстраиваются таким образом, чтобы минимизировать возможные уязвимости.

Заключение

Проведённый анализ современных технологий разработки и проектирования веб-ресурсов показывает, что веб-экосистема продолжает динамично развиваться, предлагая разработчикам

всё более разнообразные инструменты и архитектурные решения. Эволюция веба – от статических страниц к многофункциональным интерактивным системам – сопровождается не просто увеличением количества технологий, но и качественным изменением подходов к созданию веб-приложений. Значение приобретают гибкость архитектурных решений, осознанный выбор технологического стека и внимание к пользовательскому опыту, который становится центральным элементом проектирования. Современные фронтенд-технологии создают условия для построения интерфейсов, сравнимых с настольными приложениями по уровню интерактивности и скорости работы, а серверные решения обеспечивают высокую производительность, возможность масштабирования и безопасное управление данными. Всё это делает разработку многоуровневым процессом, в котором важно учитывать не только технические, но и организационные аспекты: от грамотного прототипирования до корректной настройки процессов тестирования и развертывания. Отдельное значение имеет синтез проектных и технологических решений. Практика показывает, что успешный веб-ресурс формируется там, где архитектурная схема, UX-проектирование и выбор инструментов воспринимаются как единая система, а не как набор разрозненных решений. В этом контексте особую ценность приобретают методы, позволяющие оценивать технологии с точки зрения их совместимости, гибкости и потенциала развития. Таким образом, современная веб-разработка требует комплексного подхода, сочетающего техническую экспертизу, понимание пользовательских сценариев и умение работать с быстро меняющейся технологической средой. Перспективы дальнейшего развития связаны с усилением роли реактивных архитектур, интеграцией облачных технологий, распространением микрофронтенд-структур и повышением автоматизации процессов разработки. Всё это позволяет утверждать, что будущие веб-ресурсы будут ещё более гибкими, интеллектуальными и ориентированными на глубокое взаимодействие с пользователем.

Список литературы

1. Аблаева Л. Н., Гафаров А. Р. Современные технологии разработки web-сайта для коммерческих компаний // Информационно-компьютерные технологии в экономике, образовании и социальной сфере, 2020. С. 130-138. EDN: TISXDN.
2. Букреев Д. А., Барановская В. С. Персонализация интерактивных цифровых медиа в образовательной среде // Международный студенческий научный вестник. 2023. № 2. EDN: OSCHEC.
3. Bukreiev D. A., Chorna A. V., Serdiuk I. M., Soloviev V. N. Features of the use of software and hardware of the educational process in the conditions of blended learning. Proceedings of the symposium on advances in educational technology, AET, 2020.
4. Bukreiev D., Chorny P., Kupchak E., Sender A. Features of the development of an automated educational and control complex for checking the quality of students. CEUR Workshop Proceedings, 2021.

5. Abowd G., Mynatt E. Charting past, present, and future research in ubiquitous computing // ACM Trans. Comput.-Hum. Interact. 2000. № 7(1). P. 29-58.

6. Koutrika G., Ioannidis Y. Personalizing queries based on networks of composite preferences // ACM Trans. Database Syst. 2010. № 35(2). P. 13:1-13:50.

7. Kientz J. Embedded capture and access: encouraging recording and reviewing of data in the caregiving domain // Personal Ubiquitous Comput. 2012. № 16(2). P. 209-221. EDN: OAQAUG.

ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ МОДУЛЬНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Иваненко О. А., Букреев Д. А.

*ФГБОУ ВО «Мелитопольский государственный
университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru*

Научный руководитель: Букреев Д. А.

Введение

Рост функциональной сложности цифровых продуктов и постоянное расширение пользовательских сценариев закономерно приводят к увеличению требований к архитектуре интерфейса. Сегодня пользовательский интерфейс перестает быть простой визуальной надстройкой над программной логикой и становится самостоятельным слоем, отвечающим за качество взаимодействия, удобство восприятия и эмоциональное восприятие продукта. В таких условиях особенно актуальными становятся подходы, позволяющие обеспечить гибкость, структурированность и масштабируемость интерфейсных решений. Одним из них является модульная архитектура пользовательских интерфейсов, которая предполагает построение UI как системы взаимосвязанных, но независимых компонентов.

Интерес к модульным интерфейсам объясняется несколькими факторами. Во-первых, современные цифровые продукты развиваются итеративно, а значит, архитектура должна быть устойчивой к изменению функциональности. Во-вторых, многоплатформенная среда – включая мобильные устройства, планшеты, веб-клиенты – требует единообразия и повторного использования интерфейсных элементов. В-третьих, рост числа технологий и инструментов разработки создаёт возможности для гибкого проектирования интерфейсов, но одновременно повышает требования к их согласованности. Модульный подход позволяет решить эти задачи, обеспечивая независимость компонентов, их композицию и возможность многократного использования в различных контекстах.

Особую значимость модульные интерфейсы приобретают в профессиональных и образовательных цифровых системах, где интерфейс должен быть не только удобным, но и предсказуемым, адаптивным и устойчивым к обновлениям. В таких проектах от интерфейса требуется

соблюдение дизайн-системы, согласованность визуальных элементов и отсутствие противоречий в пользовательских сценариях. Именно это делает модульность не просто техническим приёмом, а архитектурной стратегией, влияющей на качество продукта в целом.

Цель исследования – анализ современных технологий и инструментов, применяемых для реализации модульных пользовательских интерфейсов, а также изучение специфики построения таких интерфейсов в условиях разработки образовательных мобильных приложений.

Материалы и методы исследования

Материалом исследования послужили современные технологии проектирования и реализации пользовательских интерфейсов, применяемые в мобильных и кроссплатформенных приложениях. В качестве эмпирической основы использованы практические решения, включая архитектурную модель приложения, структуру модульных компонентов, особенности их взаимодействия и принципы построения интерфейсной части проекта. Дополнительно анализ опирался на документацию инструментов Jetpack Compose, Flutter, React Native и SwiftUI, а также на рекомендации по структуре дизайн-систем и компонентных библиотек, которые определяют актуальные стандарты в сфере UI-разработки.

Методологически исследование основано на сочетании теоретического анализа и сравнительного подхода. На первом этапе были изучены концептуальные источники, посвящённые компонентному проектированию, архитектурным шаблонам пользовательских интерфейсов (MVC, MVP, MVVM) и принципам модульности в интерфейсных системах. Это позволило сформировать целостное представление о фундаментальных подходах, лежащих в основе модульных UI.

На втором этапе проведён сравнительный анализ технологий, применяемых для разработки модульных интерфейсов. Сравнение осуществлялось по ряду критериев: гибкость архитектуры, поддержка композиции компонентов, удобство прототипирования, совместимость с различными платформами, требования к инфраструктуре и возможности расширения функциональности. Такой подход позволил выделить особенности нативных и кроссплатформенных инструментов, а также определить их области рационального применения.

Результаты исследования и их обсуждение

Переход к модульным пользовательским интерфейсам стал естественным ответом на усложнение цифровых систем и рост требований к качеству взаимодействия с пользователем. Современные программы – особенно мобильные и кроссплатформенные – уже не состоят из од-

ного монолитного интерфейса. Они представляют собой совокупность взаимосвязанных компонентов, которые должны быть независимыми, гибкими, легко комбинируемыми и при этом визуально согласованными. Именно эти требования формируют теоретическую основу модульности, позволяя воспринимать интерфейс как динамическую структуру, способную развиваться параллельно с функциональностью приложения. Модульный пользовательский интерфейс можно определить, как систему, построенную из самостоятельных компонентов, которые обладают четко определённой функциональностью и могут использоваться повторно в различных частях приложения. Такой подход исходит из принципа композиции: интерфейс формируется не из единой монолитной структуры, а из наборов элементов, каждый из которых отвечает за свой участок логики и визуального представления [2].

Важным преимуществом модульного UI является возможность изменять отдельные элементы без воздействия на всю систему. Это существенно повышает устойчивость интерфейса к обновлениям и сокращает трудоёмкость разработки. Модульность также способствует упорядоченности кода: разработчики работают с изолированными блоками, что облегчает тестирование, документирование и расширение функционала.

Такой подход особенно востребован в образовательных мобильных приложениях, где множество интерфейсных элементов – карточки материалов, блоки тестов, интерактивные панели – могут повторяться в различных разделах, оставаясь при этом визуально согласованными.

Компонентное проектирование базируется на нескольких ключевых принципах, определяющих структуру и поведение модульных интерфейсов.

1. Разделение ответственности.
2. Компонуемость.
3. Повторное использование.
4. Независимость и слабое зацепление.
5. Единый визуальный стиль.

Такая организация делает интерфейс одновременно гибким и устойчивым – качеством, необходимым в продуктах, ориентированных на длительное развитие.

Архитектурные модели определяют, каким образом разделяются данные, логика и представление [1]. Для модульных UI наиболее распространены MVC, MVP и MVVM.

1. MVC (Model–View–Controller) – эта модель является одной из первых попыток разделить логику и представление. Контроллер выступает посредником, управляя данными и изменениями UI. Однако в современных приложениях MVC часто оказывается слишком тесно связанной конструкцией.

2. MVP (Model–View–Presenter) – эта архитектура усиливает разделение обязанностей:

презентер полностью управляет взаимодействием между моделью и представлением. Модель получает чёткие границы ответственности, а UI становится «тонким» слоем. MVP хорошо подходит для небольших интерфейсов, но может усложнять структуру при росте приложения.

3. MVVM (Model–View–ViewModel) – двусторонняя привязка данных, позволяющая UI автоматически реагировать на изменение состояния модели. Компонент ViewModel отделяет бизнес-логику от визуального слоя, снижая связность элементов и упрощая тестирование. MVVM особенно эффективно работает при создании модульных интерфейсов в таких инструментах, как Jetpack Compose и Flutter, где декларативная модель представления organically сочетается с реактивным управлением состоянием.

Современные технологии и инструменты реализации модульных UI

Современная практика разработки пользовательских интерфейсов характеризуется появлением большого числа инструментов, ориентированных на построение интерфейса как системы взаимосвязанных модулей. Эти инструменты предлагают разные модели взаимодействия, подходы к управлению состоянием и механизмы визуального представления данных, но объединяет их одно: стремление обеспечить декларативный, устойчивый и легко расширяемый UI. Наибольшую значимость при этом приобретают нативные фреймворки нового поколения и кроссплатформенные решения, позволяющие создавать модульные компоненты, которые можно использовать на разных этапах и в разных частях приложения [3].

Появление Jetpack Compose и SwiftUI стало переломным моментом в нативной мобильной разработке. Эти технологии представляют декларативную модель построения интерфейсов, что полностью соответствует идее модульности: разработчик описывает, каким должен быть компонент, а фреймворк самостоятельно управляет его состоянием, деревом элементов и процессом перерисовки.

Jetpack Compose предлагает модель интерфейса, построенную на функциях, которые описывают UI-компоненты. Каждый компонент становится изолированным модулем, который может быть использован в любом месте приложения. Благодаря механизму recomposition интерфейс автоматически обновляется при изменении состояния, а использование ViewModel обеспечивает слабую связность между слоями.

Сильные стороны Compose:

- минимизация шаблонного кода;
- естественная поддержка MVVM;
- удобство создания комплексных UI-элементов;
- гибкость композиции.

SwiftUI – аналогичная декларативная технология для iOS. Она использует структурный подход, где интерфейс описывается через комбинацию структурных элементов.

Преимущества SwiftUI:

- простой способ описания интерфейса;
- высокая согласованность внешнего вида;
- глубокая интеграция с экосистемой Apple;
- возможность комбинирования модулей в иерархию.

Обе технологии создают среду, где модульность – не дополнительная опция, а встроенный принцип, что делает их идеальными для разработки современных интерфейсов [4].

Кроссплатформенные решения позволяют создавать единый UI-код, который работает на разных платформах – Android, iOS, Web, Desktop. В контексте модульных интерфейсов такие технологии оказываются особенно важными, поскольку позволяют переиспользовать компоненты между проектами и платформами.

Flutter предлагает декларативную модель построения интерфейса и использует виджеты как основную единицу UI. Виджет в Flutter по своей сути является модулем: он обладает собственным состоянием и может быть встроен в другие компоненты. Благодаря этому UI легко структурируется в дерево виджетов, что обеспечивает высокую гибкость композиции.

Особенности Flutter, важные для модульности:

- единая система виджетов;
- чёткое разделение Stateless и Stateful компонентов;
- широкие возможности кастомизации;
- высокая согласованность UI на разных платформах.

React Native применяет модель React – компонентный подход, построенный на JavaScript и JSX. Каждый компонент представляет собой отдельный модуль, который может быть многократно использован в разных экранах. Это совпадает с принципами Atomic Design и позволяет создавать структуры высокой сложности.

Kivu менее распространён, он активно используется в образовательных и экспериментальных проектах. Его компонентная архитектура и механизмы разметки позволяют строить гибкие интерфейсы, где модульность поддерживается за счёт файловой структуры и чёткого разделения логики и представления. Кроссплатформенные технологии усиливают идею модульности, позволяя уменьшить дублирование кода и поддерживать единый стиль взаимодействия на разных устройствах [5]. Проектирование модульных интерфейсов не ограничивается только кодом. На этапе концептуальной разработки важную роль играют инструменты прототипирования и дизайн-системы. Figma, Adobe XD и другие инструменты позволяют:

- создавать интерфейсы в виде архитектуры компонентов;

- выделять повторяющиеся элементы в отдельные UI-компоненты;

- управлять стилями, сетками, типографикой;
- формировать дизайн-системы, используемые в кодовой базе.

Дизайн-система становится фундаментом модульности: она определяет правила построения интерфейсов, внешний вид компонентов, их состояние и взаимодействие. На основе дизайн-системы разрабатываются компонентные библиотеки, которые затем используются как в прототипировании, так и в программной реализации. Таким образом, связь между дизайнерскими и техническими инструментами обеспечивает полноценный цикл создания модульного UI – от визуального моделирования до интеграции в кодовую архитектуру.

Особенности реализации модульных UI в образовательных цифровых продуктах

Образовательные мобильные приложения предъявляют особые требования к пользовательскому интерфейсу. Он должен быть понятным для широкого круга пользователей, адаптивным, устойчивым к изменениям и способным поддерживать многокомпонентную структуру содержания. В таких условиях модульный UI становится не просто удобным архитектурным подходом, а фактически необходимым инструментом, обеспечивающим длительное развитие продукта и сохранение качества взаимодействия.

Особенность образовательных систем заключается в том, что интерфейс должен не только отображать информацию, но и поддерживать активное обучение: вовлекать пользователя, обеспечивать последовательность этапов, визуализировать прогресс и соответствовать методическим задачам. Всё это требует использования гибких интерфейсных конструкций, которые легко адаптируются к содержанию и различным контекстам использования.

В отличие от коммерческих приложений, где дизайн может быть ориентирован на маркетинговые задачи, образовательные продукты предъявляют более строгие требования к функциональной ясности и структурированности.

Среди таких требований можно выделить:

1. Понятность и когнитивная доступность.
2. Предсказуемость взаимодействия.
3. Адаптивность к содержанию.
4. Поддержка динамики обучения.
5. Устойчивость и масштабируемость.

Образовательные платформы часто расширяются: добавляются курсы, модули, уровни сложности. Это делает модульный подход единственно рациональным. Таким образом, интерфейс образовательного приложения должен быть не только эстетичным, но и функционально надёжным, что возможно только при грамотной модульной организации UI.

В образовательных приложениях используются устойчивые паттерны интерфейса, которые зарекомендовали себя как эффективные для восприятия и навигации. Среди них:

1. Карточечные компоненты, применяемые для представления единиц контента – уроков, заданий, тем.
2. Интерактивные блоки, управляющие ответами, выбором вариантов, переходом по шагам.
3. Вспомогательные панели прогресса, визуализирующие результаты обучения и стимулирующие мотивацию.
4. Модульные списочные структуры, адаптирующиеся под различный объём данных.
5. Диалоговые окна и подсказки, которые помогают ориентироваться в интерфейсе и выполнять задания.

Все эти паттерны отлично сочетаются с модульной архитектурой, поскольку каждый представляет собой отдельный компонент с собственным состоянием и поведением. Их можно многократно использовать на разных экранах и в разных сценариях [7].

Каждый компонент реализован таким образом, чтобы его можно было использовать в других частях интерфейса, модифицировать или расширять без переписывания логики приложения. Именно такая гибкость делает модульный UI особенно ценным в образовательных цифровых продуктах.

Практическая реализация модульного пользовательского интерфейса предполагает не просто применение отдельных компонентов, а осознанное построение архитектуры, в которой каждый элемент занимает своё место и взаимодействует с другими через чётко определённые механизмы [6]. В условиях реальных проектов модульность проявляется не только в структуре отдельных экранов, но и в общей системе управления состоянием, способах интеграции и принципах поддержки дизайн-системы. Именно эти аспекты определяют жизнеспособность интерфейса при изменении требований, масштабировании или добавлении новых функциональных блоков.

Модульная архитектура позволяет организовать разработку таким образом, чтобы каждая часть интерфейса могла развиваться независимо, при этом оставаясь частью целостной системы. На практике такой подход в значительной степени снижает связанность кода, улучшает тестируемость компонентов и обеспечивает единообразие визуальной среды.

Одним из наиболее эффективных способов структурирования модульных интерфейсов является сочетание архитектуры MVVM и концепции Atomic Design. Обе методики дополняют друг друга, обеспечивая как структурную ясность, так и визуальную последовательность.

MVVM обеспечивает организацию логики интерфейса через разделение модели, представ-

ления и слоя ViewModel. В рамках модульного UI это особенно важно:

- каждый компонент получает собственную ViewModel или часть общей модели;
- состояние компонента изолировано и не влияет на работу других модулей;
- обновления происходят автоматически благодаря реактивной модели данных.

Это делает интерфейс устойчивым к модификациям и значительно облегчает его тестирование. Выбор конкретного механизма зависит от сложности интерфейса, уровня вложенности модулей и требований к реактивности. Во всех случаях цель остаётся неизменной: обеспечить согласованность модулей при сохранении их автономности.

Заключение

Проведённое исследование показало, что модульные пользовательские интерфейсы становятся ключевым направлением развития современных цифровых систем, обеспечивая структурированность, гибкость и устойчивость интерфейсной архитектуры. Применение компонентного подхода позволяет создавать интерфейсы, которые легко адаптируются к расширению функциональных возможностей, поддерживают единообразие визуального стиля и остаются удобными для сопровождения на протяжении всего жизненного цикла приложения. Анализ теоретических основ модульности продемонстрировал, что архитектурные модели MVC, MVP и особенно MVVM обеспечивают надёжный механизм разделения ответственности между слоями интерфейса. В сочетании с концепцией Atomic Design они формируют фундамент для построения UI-систем, ориентированных на повторное использование и независимость компонентов. Такое объединение архитектурного и визуального подходов позволяет упорядочивать структуру интерфейса и увеличивать его предсказуемость. Рассмотрение современных инструментов – Jetpack Compose, SwiftUI, Flutter, React Native – показало, что именно декларативные технологии предоставляют наиболее гармоничную среду для реализации модульных интерфейсов. Эти фреймворки позволяют разработчику сосредоточиться на логике и назначении компонентов, а управление состоянием и рендерингом берут на себя встроенные механизмы платформы. Дополнение разработки инструментами прототипирования, включая Figma и другие дизайн-системы, обеспечивает согласованность визуального языка и облегчает коммуникацию между разработчиками и дизайнерами.

Список литературы

1. Букреев Д. А., Барановская В. С. Персонализация интерактивных цифровых медиа в образовательной среде // Международный студенческий научный вестник. 2023. № 2.

2. Букреев Д. А., Луц С. М. Особенности разработки web-платформ автоматизированного взаимодействия с клиентом // Современные проблемы геометрического моделирования и информационные технологии: материалы II Межрегиональной научно-практической конференции преподавателей и студентов, посвященной 60-летию образования Мелитопольской школы прикладной геометрии (Мелитополь, 28 мая 2024 года). Мелитополь: Мелитопольский государственный университет, 2024. С. 76-85. EDN: BVUZYR.

3. ISO9241. Ergonomics of human-system interaction. URL: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (дата обращения: 12.12.2025).

4. Lazuardi M. L., Sukoco I. Design Thinking, David Kelley, Tim Brown: Otak Dibalik Penciptaan Aplikasi Gojek // Organum: Jurnal Saintifik Manajemen Dan Akuntansi. 2019. № 2.

5. Muslim E., Moch B. N., Wilgert Y., Utami F. F., Indriyani D. User interface redesign of e-commerce platform mobile application (Kudo) through user experience evaluation to increase user attraction. IOP Conference Series: Materials Science and Engineering. 2019. № 508(1). P. 012113.

6. Song's M. Z. Design and implementation of a vue. js-based college teaching system. 2019.

7. Sultan M. Angular and the Trending Frameworks of Mobile and Web-based Platform Technologies: A Comparative Analysis // Proc. Future Technologies Conference. 2017. С. 928-936.

ИСПОЛЬЗОВАНИЕ ОНТОЛОГИЙ В ПОСТРОЕНИИ Е-ПРОФИЛЯ СТУДЕНТА

Иванив О. С., Строкань О. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: oksanaivaniv45@gmail.com*

Научный руководитель: Строкань О. В.

Введение

В условиях стремительной цифровизации объём создаваемых данных растёт настолько быстро, что традиционные подходы к их обработке и анализу теряют эффективность. Ещё в конце XX века исследователи прогнозировали экспоненциальный рост информационных массивов, и в настоящее время этот прогноз реализуется в полной мере: скорость генерации данных ежегодно увеличивается, при этом значительная их часть относится к неструктурированной информации, требующей новых методов интерпретации и осмысления [1].

Данные процессы оказывают непосредственное влияние на рынок труда. Согласно аналитическим обзорам, специалисты, работающие с анализом и интерпретацией данных, входят в число наиболее востребованных профессионалов в IT-сфере [2]. От них ожидается не только владение инструментами обработки информации, но и способность выявлять взаимосвязи, формировать обоснованные выводы и принимать решения на основе комплексного анализа данных [3]. Повышенный спрос на такие компетенции отражается и в уровне заработных плат, который стабильно превышает среднерыночные показатели [4].

Рост объёмов данных затрагивает и систему высшего образования. Так, современный студент должен уметь ориентироваться в цифровой сре-

де, осваивать новые образовательные форматы и формировать индивидуальную образовательную траекторию. В этих условиях особую значимость приобретает задача структурирования, интеграции и аналитического использования данных о ходе обучения и развитии студентов.

Параллельно трансформируются модели взаимодействия между университетами, государством и работодателями. Ответственность за профессиональную траекторию всё в большей степени возлагается на самого обучающегося, тогда как государственные структуры и бизнес заинтересованы в раннем выявлении и поддержке перспективных кадров. Это формирует запрос на инструменты, способные объединять сведения о компетенциях студентов, динамике их обучения и потребностях рынка труда в единую аналитическую систему.

В данной статье рассматривается онтологический подход как методологическая основа построения электронного профиля студента.

Целью работы является анализ сущности онтологий и семантических технологий, обоснование их применения в образовательных цифровых системах, а также выявление возможностей и ограничений использования онтологий при формировании е-профиля студента.

В работе не только теоретически обосновывается применение онтологий в образовательных цифровых системах, но и предлагается обобщённая онтологическая схема структуры е-профиля студента, иллюстрирующая интеграцию разнородных данных в единую семантическую модель.

Материал и методы исследования

Материалами исследования послужили научные труды, посвящённые онтологическому подходу, семантическим технологиям и цифровизации образовательных процессов [5, 6, 9, 10], а также публикации, рассматривающие вопросы формирования электронных профилей студентов [7] и анализа образовательных траекторий. Дополнительно использовались открытые методические и аналитические материалы в сфере высшего образования, отражающие современные требования к оценке образовательных результатов и компетенций обучающихся.

Методологическую основу исследования составляют системный и онтологический подходы, позволяющие рассматривать е-профиль студента как семантически согласованную модель, объединяющую разнородные данные об учебной, научной, проектной и внеучебной деятельности. В рамках системного подхода электронный профиль анализируется как элемент более широкой экосистемы взаимодействия университета, государства и рынка труда, что соответствует логике, изложенной во введении и выводах статьи.

В исследовании применялись методы концептуального и логического анализа, направленные

ные на выделение ключевых сущностей предметной области и определение смысловых связей между ними. Основным методом выступает онтологическое моделирование, включающее формализацию понятий, отношений и атрибутов, характеризующих образовательную траекторию студента.

Результаты исследования и их обсуждение

Онтология как научное понятие имеет длительную историю и изначально формировалась в рамках философии как учение о сущем и фундаментальных принципах бытия [8]. В классическом понимании онтология была направлена на исследование универсальных категорий и закономерностей существования мира. В условиях развития информационных технологий содержание этого понятия существенно изменилось.

В компьютерных науках под онтологией понимают формализованную модель знаний, описывающую ключевые сущности предметной области, их свойства и взаимосвязи. Иначе говоря, онтология фиксирует, какие объекты существуют в системе и каким образом они логически соотносятся друг с другом, что позволяет создавать концептуальное представление предметной области в форме, доступной для машинной обработки [9].

Развитие онтологических моделей тесно связано с задачами построения интеллектуальных информационных систем. Для решения аналитических и управленческих задач компьютеру недостаточно оперировать формальными данными — требуется возможность работать со знаниями. Это предполагает перенос в цифровую среду тех структур, которыми человек пользуется в процессе мышления: понятий, отношений, логических правил и механизмов рассуждений. Формализация таких структур создаёт основу для логического вывода, интерпретации информации и поддержки принятия решений.

Данный переход тесно связан с понятием семантики. В информационных системах семантика отражает смысл информации и её соотносённость с объектами реального мира. Для вычислительной машины данные представляют собой лишь формальные значения, не обладающие смыслом сами по себе [9]. Онтологии выступают ключевым инструментом семантического моделирования, поскольку позволяют формализовать знания таким образом, чтобы смысловая структура предметной области стала доступной для машинной интерпретации. Именно это отличает семантические технологии от традиционных ИТ и обеспечивает переход от обработки данных к работе со знаниями.

Практическая значимость онтологического и семантического подходов наиболее наглядно проявляется в сфере образования. Онтологии позволяют структурировать учебные матери-

алы, дисциплины, компетенции и результаты обучения, формируя единую семантическую образовательную среду [10]. Это создаёт возможности для автоматизированного сопоставления учебных целей, формирования индивидуальных образовательных траекторий, подбора контента в соответствии с уровнем подготовки обучающихся и выявления пробелов в знаниях.

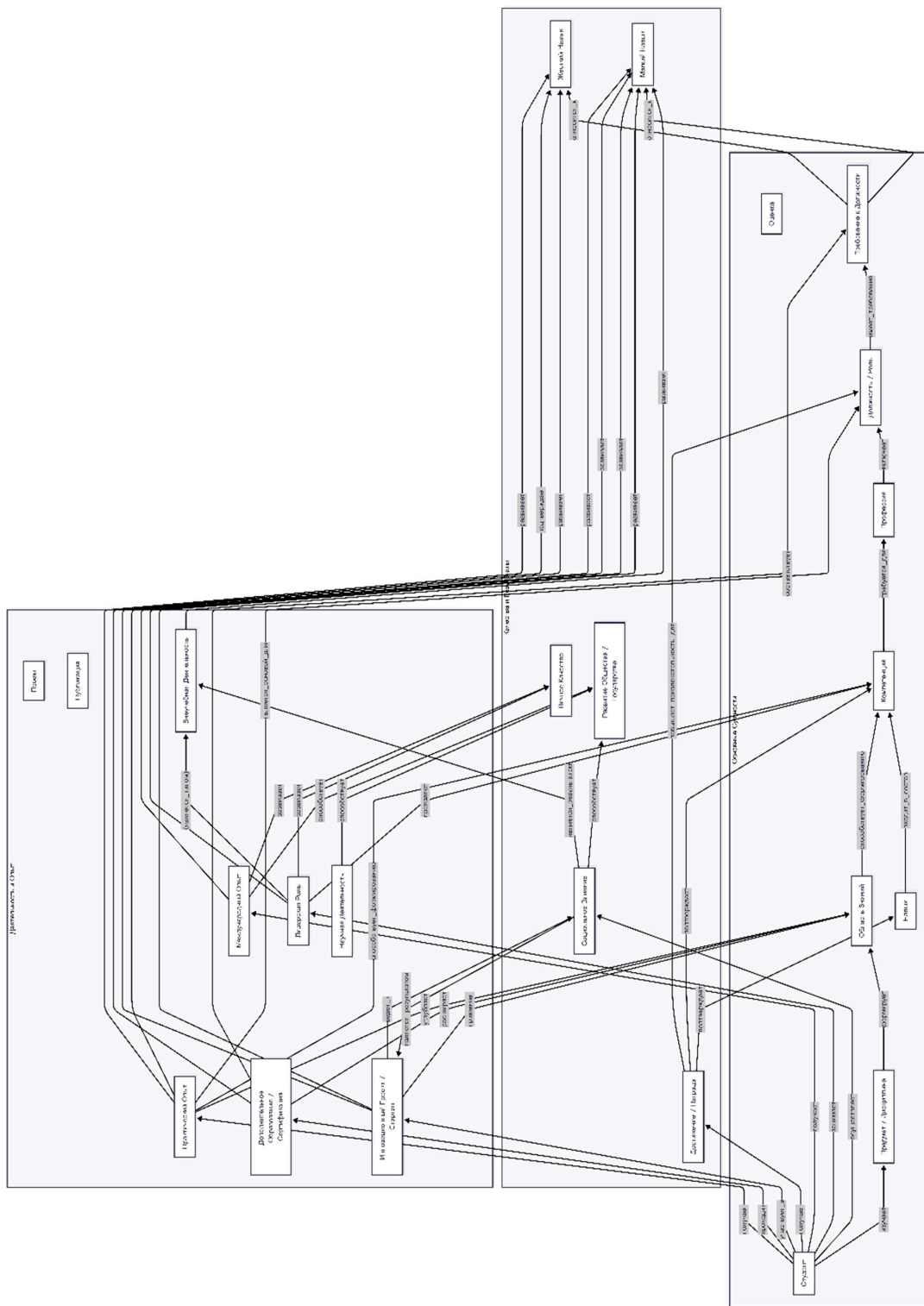
Кроме того, онтологические модели можно использовать для интеграции разрозненных цифровых образовательных ресурсов как связующим семантическим слоем между системами управления обучением, цифровыми библиотеками, профессиональными стандартами и платформами оценки компетенций. Такое объединение обеспечивает согласованность данных и формирует основу для интеллектуальных сервисов анализа и поддержки образовательного процесса.

В рамках электронного профиля студента онтологический подход позволяет объединить разнородные сведения в единую смысловую модель.

Для наглядного представления структуры е-профиля студента в рамках данного исследования была разработана онтологическая схема, отражающая основные сущности и связи между ними. На рисунке показано, каким образом учебная, научная, проектная, общественная и внеучебная деятельность студента объединяются в единую семантическую модель. Ключевым преимуществом онтологий является способность устанавливать логические связи между элементами, которые на первый взгляд не связаны напрямую, формируя целостное представление об образовательной траектории и потенциале студента. Так, представленная схема иллюстрирует данный принцип интеграции разнородных данных и может служить основой для дальнейшей реализации онтологически ориентированных образовательных систем.

Схема демонстрирует ключевые сущности (концепции) и отношения между ними, формирующие комплексный электронный профиль студента. Модель включает академические успехи, участие в научной и внеучебной деятельности, практический и международный опыт, развитие жестких и мягких навыков, формирование компетенций, а также их влияние на профессиональное развитие и вклад в общественное благо.

Современная образовательная среда отличается высокой вариативностью и гибкостью. Государство активно расширяет инструменты поддержки студентов, включая грантовые программы, конкурсы, академическую мобильность, научные инициативы и социальные меры. Эти механизмы отражают заинтересованность государства в развитии качественного кадрового потенциала, однако их эффективность может быть объективно оценена только при наличии целостной картины образовательного развития обучающихся.



Онтологическая модель e-профиля студента

Онтологически организованный е-профиль студента позволяет агрегировать данные из множества источников и анализировать не отдельные показатели, а комплексные траектории развития. Это даёт возможность выявлять сильные и слабые стороны подготовки, оценивать влияние различных образовательных инструментов и принимать обоснованные управленческие решения.

Для бизнеса структурированная информация о студентах имеет принципиальное значение, поскольку качество подготовки специалистов и уровень их компетентности должны соответствовать требованиям современной экономики. В условиях гибких образовательных моделей стандартные показатели успеваемости теряют информативность. Онтологический е-профиль предоставляет работодателям более полное представление о реальных навыках и потенциале выпускников, облегчая отбор кадров и планирование их профессионального развития.

На макроуровне формирование онтологически согласованных е-профилей студентов способствует построению аналогичного профиля образовательной организации. Это позволяет государственным органам анализировать эффективность образовательных программ, оценивать результативность мер поддержки и планировать развитие системы высшего образования, стимулируя вузы к комплексному развитию всех сфер студенческой жизни.

Несмотря на широкие возможности онтологического подхода, при интерпретации е-профиля важно учитывать ряд факторов, связанных с особенностями человеческого поведения и ограничениями исходных данных.

Во-первых, формальные показатели не всегда отражают реальный уровень знаний и компетенций. Высокие оценки или активное участие в мероприятиях могут быть результатом внешних факторов и не гарантируют глубины усвоения материала. Во-вторых, онтологии фиксируют факт участия, но не качество вклада, что требует дополнения данных экспертными оценками и портфолио работ.

Кроме того, различия в жизненных условиях студентов и возможная неполнота цифровых данных могут влиять на интерпретацию профиля. Онтологическая модель корректно отражает структуру информации, однако её аналитическая ценность напрямую зависит от полноты и достоверности исходных данных.

Заключение

В статье рассмотрен онтологический подход как методологическая основа построения электронного профиля студента. Показано, что использование онтологий и семантических технологий позволяет перейти от разрозненных

данных к целостным моделям знаний, обеспечивающим более глубокий анализ образовательных траекторий.

Применение онтологий в образовательных цифровых системах способствует повышению прозрачности и управляемости процессов обучения, укрепляет взаимодействие между государством, университетами и бизнесом, а также создаёт условия для более точной и адресной подготовки специалистов. Несмотря на необходимость учёта человеческого фактора и ограничений данных, онтологический е-профиль представляет собой перспективный инструмент развития современной системы высшего образования.

Разработанная в рамках исследования онтологическая схема структуры е-профиля студента подтверждает практическую применимость предложенного подхода и может быть использована в качестве концептуальной основы для проектирования образовательных аналитических систем.

Список литературы

1. Abbasov A. M. Information Boom: New Trends and Expectations // *Soft Computing: State of the Art Theory and Novel Applications* / eds. Yager R., Abbasov A., Reformat M., Shahbazova S. *Studies in Fuzziness and Soft Computing*. Vol. 291. Berlin, Heidelberg: Springer, 2013. DOI: 10.1007/978-3-642-34922-5_1.
2. Аналитик данных: перспективы и важность профессии // *Молодой ученый*. 2020. № 14 (304). [Электронный ресурс]. URL: <https://moluch.ru/archive/304/105176> (дата обращения: 18.12.2025).
3. Data analytics developers jobs market: July 2025 // CHU.ST. 2025. [Электронный ресурс]. URL: <https://chu.st/analytics/data-analytics-developers-jobs-market-july-2025/> (дата обращения: 18.12.2025).
4. Зарплаты аналитиков данных разного уровня в России: исследование Skillbox Media // Skillbox Media. [Электронный ресурс]. URL: <https://skillbox.ru/media/code/zarplaty-analitikov-dannykh-raznogo-urovnya-v-rossii-issledovanie-skillbox-media/> (дата обращения: 18.12.2025).
5. Лапшин В.А. Онтологии в компьютерных системах. М.: Научный мир, 2010.
6. Gruber T.R. The role of common ontology in achieving sharable, reusable knowledge bases // *Principles of Knowledge Representation and Reasoning. Proceedings of the Second International Conference* / J. A. Allen, R. Fikes, E. Sandewell (eds.). Morgan Kaufmann, 1991. P. 601–602.
7. Аликина Е. В., Мальцев Д. В. Оптимизация электронного портфолио студента многопрофильного вуза // *Вестник Томского государственного университета*. 2023. № 488. С. 14–22. DOI: 10.17223/15617793/488/2.
8. Плешкова А. Ю. Онтологии в управлении образовательным процессом // *Онтология проектирования*. 2022. Т. 12. № 4 (46). С. 506–517. DOI: 10.18287/2223-9537-2022-12-4-506-517.
9. Introduction to Semantic Technologies. TriniData. [Электронный ресурс]. URL: <https://trinidata.ru/files/SemanticIntro.pdf> (дата обращения: 18.12.2025).
10. Смирнова Е.В., Добрица Е.К., Демиденко Н.О. Использование онтологий в образовательных процессах // *Проблемы науки*. 2017. № 22 (104). [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/ispolzovanie-ontologiy-v-obrazovatelnyh-protsessah> (дата обращения: 18.12.2025).

МЕТОДЫ И МОДЕЛИ ПРОГНОЗИРОВАНИЯ СПОРТИВНЫХ СОБЫТИЙ

Константинов А. В., Букреев Д. А.

*ФГБОУ ВО «Мелитопольский государственный
университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru*

Научный руководитель: Букреев Д. А.

Введение

Современное развитие цифровых технологий привело к значительному увеличению объёмов спортивной статистики, доступной для анализа, что существенно изменило характер спортивной аналитики и расширило возможности прогнозирования игровых событий. В условиях, когда практически каждый матч сопровождается подробными числовыми и видеоданными, а информационные платформы регулярно обновляют показатели команд и игроков, возрастает необходимость применения интеллектуальных методов для анализа этих данных и построения достоверных прогнозов. В частности, футбол как один из наиболее популярных и статистически насыщенных видов спорта предоставляет исследователям уникальные возможности для выявления скрытых закономерностей и моделирования вероятности спортивных исходов. Традиционные экспертные подходы постепенно уступают место методам статистического анализа, машинного обучения и нейросетевого моделирования. Эти подходы позволяют учитывать широкий спектр факторов – от индивидуальных игровых показателей до трендовых характеристик команд, событий матча и контекстных параметров. Развитие методов искусственного интеллекта, доступность инструментов обработки данных и рост вычислительных мощностей способствуют тому, что прогнозирование спортивных событий приобретает точный, формализованный и воспроизводимый характер. При этом особую значимость приобретает разработка программных решений, способных объединять сбор данных, их подготовку, обучение моделей и визуализацию результатов в единой функциональной системе.

Актуальность данного исследования определяется необходимостью повышения точности прогнозирования и упрощения процедуры аналитической обработки для широкого круга пользователей – от спортивных аналитиков и преподавателей до любителей спорта. Несмотря на существование зарубежных сервисов спортивной аналитики, большинство из них ориентированы на крупные коммерческие платформы и не учитывают специфики локальных чемпионатов или учебно-исследовательских задач. В этой связи представляется важным создание программного инструмента, реализующего современные математические и алгоритмические

подходы, но при этом доступного, прозрачного и адаптируемого под конкретные требования.

Цель исследования – анализ методов и моделей прогнозирования спортивных событий и разработка программного средства, реализующего интеллектуальный подход к предсказанию результатов футбольных матчей.

Материалы и методы исследования

Материалом исследования выступили статистические данные о футбольных матчах, включающие как базовые показатели команд, так и расширенные характеристики, отражающие их игровую форму, результативность и поведенческие особенности в конкретных соревнованиях. Используемый набор данных был сформирован на основе открытых спортивных ресурсов, агрегаторов статистики и архивов матчей, что позволило получить достаточно репрезентативную выборку для построения моделей прогнозирования.

Первоначальный этап исследования был связан с предварительной обработкой данных. Статистические записи зачастую содержали пропуски, дубли и некорректные значения, поэтому была проведена очистка выборки, унификация форматов и нормализация числовых признаков.

Для выбора наиболее значимых факторов использовались методы корреляционного анализа, статистической значимости признаков и оценка важности атрибутов с помощью алгоритмов машинного обучения, таких как Random Forest и Gradient Boosting. Обучение моделей проводилось на разделённых обучающих и тестовых выборках, что обеспечивало объективную оценку качества прогнозирования. Для оценки эффективности применялись стандартные метрики классификации – accuracy, precision, recall, F1-score, а также показатели качества вероятностных предсказаний, включая log-loss и ROC-AUC. Подход с перекрёстной валидацией позволил минимизировать эффект случайных флуктуаций и повысить достоверность результатов.

Результаты исследования и их обсуждение

Прогнозирование спортивных событий представляет собой сложную аналитическую задачу, требующую учёта большого количества факторов, различающихся по природе, стабильности и степени влияния на итоговый результат. В отличие от классических областей статистического моделирования, где параметры могут быть относительно устойчивыми во времени, спортивные данные характеризуются высокой динамичностью: состояние команды, тактическая схема, психологические аспекты, мотивация и даже внешние условия способны существенно изменять исход события. Именно поэтому выбор методов прогнозирования требует

комплексного подхода, основанного на сочетании математических, статистических и интеллектуальных моделей.

Статистические методы традиционно являются основой спортивной аналитики благодаря своей интерпретируемости и простоте применения. Наиболее распространённой моделью является распределение Пуассона, которое применяется для прогнозирования количества голов в футбольном матче [1]. Модель исходит из предположения, что количество забитых голов можно описать как редкое событие, происходящее в рамках определённого среднего значения. Это позволяет оценивать вероятности различных счётов и, следовательно, вероятность победы, ничьей или поражения. Другим важным статистическим инструментом является логистическая регрессия, применяемая для оценки вероятности наступления бинарных или многоклассовых исходов [2]. Её преимущество заключается в возможности учёта большого количества независимых переменных и получении вероятностной интерпретации результатов. Однако статистические модели имеют ограниченную способность к работе со сложными нелинейными зависимостями и могут терять точность при наличии значительных шумов в данных или высоком уровне вариативности спортивных показателей.

Байесовские подходы основываются на теории вероятностей и позволяют учитывать априорные знания, постепенно корректируя прогнозы по мере появления новых данных. В спортивной аналитике байесовские сети применяются для моделирования взаимодействий между факторами и построения гибких прогнозов, опирающихся на вероятностное распределение параметров. Преимущество этого подхода проявляется при анализе малых выборок или в условиях, когда параметры системы меняются неравномерно. Однако высокая вычислительная сложность и потребность в точной настройке структуры моделей ограничивают их практическое использование в оперативных системах прогнозирования.

Методы машинного обучения получили широкое распространение благодаря способности выявлять сложные взаимосвязи в данных, которые могут быть недоступны классическим статистическим моделям. Среди наиболее эффективных алгоритмов, используемых в спортивной аналитике, можно выделить:

- деревья решений, позволяющие выявлять правила, определяющие исход матча на основе пороговых разделений данных;
- Random Forest, представляющий собой ансамбль деревьев решений и обеспечивающий высокую устойчивость к шумам и переобучению;
- градиентный бустинг, который последовательно улучшает прогнозы, концентрируясь на наиболее сложных для классификации случаях;

- метод опорных векторов (SVM), эффективно работающий на данных высокой размерности и обеспечивающий чёткие разделяющие поверхности между классами.

Эти методы обладают высокой обобщающей способностью и позволяют формировать более стабильные модели, особенно при правильной настройке параметров и достаточном объёме тренировочной выборки.

Применение нейронных сетей в спортивной аналитике стало возможным благодаря росту вычислительных мощностей и доступности больших массивов данных. В отличие от статистических методов, нейросети способны автоматически выявлять скрытые закономерности, используя многослойные представления данных. В рамках исследования рассматривались полносвязные нейронные сети, а также более сложные архитектуры, применяемые для анализа временных рядов, такие как LSTM и GRU. Последние позволяют учитывать динамику формы команды, последовательность предыдущих результатов и ряд других временных факторов, которые существенно влияют на вероятность победы.

Нейросетевые модели часто демонстрируют более высокую точность по сравнению с классическими методами, однако требуют тщательно подготовленных данных и значительных вычислительных ресурсов для обучения.

Особое место среди современных подходов занимают ансамблевые и гибридные модели, комбинирующие преимущества различных методов. Их эффективность объясняется тем, что каждый алгоритм выделяет свои характерные закономерности, а объединение результатов позволяет компенсировать индивидуальные недостатки.

В спортивной аналитике наибольшее распространение получили ансамбли моделей, объединяющие:

- статистические методы и ML;
- ML и нейросетевые модели;
- несколько моделей машинного обучения с разной структурой.

Такие ансамбли позволяют получать более устойчивые прогнозы, особенно на выборках с высокой вариативностью.

Каждый подход имеет свои преимущества и ограничения, а их выбор определяется целями исследования, качеством данных и особенностями конкретного спортивного соревнования.

Модели машинного обучения в прогнозировании результатов матчей

Модели машинного обучения занимают центральное место в современном прогнозировании спортивных событий благодаря способности выявлять сложные и неочевидные взаимосвязи между игровыми показателями. В отличие от классических статистических методов, ориентированных преимущественно на линейные зависимости, алгоритмы машинного обучения

позволяют работать с высокоразмерными, шумными и неоднородными данными, что делает их особенно полезными для анализа футбольных матчей, где результат определяется сочетанием множества факторов. Применение таких моделей требует тщательной подготовки данных, корректного выбора признаков и взвешенного подхода к организации процесса обучения. Качество обучающей выборки является ключевым фактором, определяющим итоговую эффективность моделей машинного обучения. Рабочий датасет содержал информацию о большом количестве футбольных матчей, включающую показатели результативности, форму команд, статистику домашних и выездных игр, а также различные контекстные характеристики: турнир, этап чемпионата, наличие значимых игровых событий и кадровых изменений в составе. Поскольку данные имели различную природу, важным этапом стала их унификация: числовые признаки нормировались, категориальные подвергались кодированию, а показатели, имеющие временную динамику, дополнялись производными характеристиками. Это позволило усилить информативность исходных данных и придать моделям способность улавливать устойчивые тренды. Для предотвращения смещения результатов выборка была разделена на тренировочную и тестовую части, а при необходимости применялась перекрёстная валидация.

Одной из ключевых задач при построении моделей стало определение наиболее значимых факторов, влияющих на исход матча. Для решения этой задачи применялись методы оценки важности признаков, основанные на ансамблевых моделях, таких как Random Forest и Gradient Boosting. Эти алгоритмы позволяют анализировать вклад каждого параметра в процесс классификации и строить ранжированные списки факторов.

По результатам анализа были выделены наиболее значимые признаки:

- текущая форма команды (результаты последних матчей);
- разница забитых и пропущенных голов;
- статистика домашних и выездных встреч;
- характеристики соперника;
- средняя результативность команды за несколько предыдущих туров;
- очковый баланс и турнирная мотивация.

Эти показатели отражают реальную игровую силу команды и обладают высокой прогностической ценностью, что подтверждается их регулярным появлением среди значимых факторов при обучении моделей [5].

В качестве основных моделей прогнозирования рассматривались алгоритмы, традиционно демонстрирующие высокую устойчивость при работе с табличными данными:

1. Деревья решений – обеспечивают интерпретируемость и быстрый процесс обучения,

но склонны к переобучению. Применялись как базовый ориентир.

2. Random Forest – представляет собой ансамбль деревьев решений и отличается способностью обобщать данные при наличии шумов. Модель показала высокую устойчивость и хорошие значения точности.

3. Gradient Boosting – один из наиболее эффективных алгоритмов для задач классификации спортивных исходов. Позволяет уточнять прогнозы путём последовательного исправления ошибок предыдущих моделей. Демонстрировал наиболее высокие значения точности и F1-метрики.

4. Метод опорных векторов – использовался для построения разделяющих поверхностей между классами, обеспечивая высокую точность, но требуя тщательной параметризации.

Все модели обучались на подготовленном датасете с использованием оптимальных процедур настройки гиперпараметров: перебора по сетке, кросс-валидации и контроля переобучения. Для повышения эффективности обучения применялись:

- функции активации ReLU;
- алгоритм оптимизации Adam;
- механизмы регуляризации (Dropout);
- нормализация входных данных.

Применение нейронной сети позволило улучшить качество прогнозов в ситуациях, когда между признаками наблюдалась выраженная нелинейная зависимость. В частности, модель лучше справлялась с прогнозированием исходов матчей между командами сопоставимого уровня, где традиционные методы часто демонстрируют погрешности.

Важно подчеркнуть, что каждая модель продемонстрировала свои преимущества в зависимости от структуры данных. Статистические методы обеспечивали хорошую интерпретируемость, модели машинного обучения – высокую точность и устойчивость, а нейронные сети – способность выявлять скрытые зависимости. Сравнение результатов показало, что наиболее эффективными оказались ансамблевые модели и нейронные сети, обладающие лучшим балансом метрик accuracy и F1-score. При этом итоговая точность предсказаний превышала показатели, характерные для классических статистических методов.

Сравнительный анализ точности различных моделей

Сравнительный анализ различных моделей прогнозирования является ключевым элементом исследования, поскольку позволяет объективно оценить, насколько успешно тот или иной алгоритм справляется с задачей классификации исходов футбольных матчей. Поскольку данные обладают сложной структурой, включают в себя как числовые, так и категориальные признаки, а также отличаются высокой вариативностью, вы-

бор наиболее эффективной модели не может быть основан на формальном сравнении отдельных метрик [3]. Требуется комплексная оценка, учитывающая устойчивость модели, точность прогнозов, способность обобщать результаты и адекватно реагировать на изменения данных [4].

Для обеспечения объективности сравнения применялся единый набор входных данных и идентичные условия разбиения выборки на тренировочную и тестовую части. Каждая модель проходила обучение с использованием одинаковых признаков, что позволяло минимизировать влияние сторонних факторов [6]. Для оценки качества использовались несколько метрик, каждая из которых отражала свою сторону эффективности:

- Accuracy – общая доля правильных предсказаний;
- Precision – точность распознавания конкретного исхода;
- Recall – полнота выявления данного класса;
- F1-score – гармоническое среднее между precision и recall;
- ROC-AUC – способность модели различать классы при разных порогах решения.

Использование нескольких метрик позволило избежать односторонней оценки и получить более полное представление о работе каждой модели. Такой подход особенно важен для прогнозирования исходов спортивных матчей, где распределение классов часто бывает несбалансированным.

1. Статистические модели показали ожидаемо стабильные, но умеренные результаты.

2. Логистическая регрессия продемонстрировала достаточно высокую точность на выборках с выраженной разницей в игровых показателях команд, однако её эффективность снижалась при анализе матчей сопоставимого уровня. Точность модели составила около 65–68 %, что соответствует средним значениям для традиционных методов спортивной аналитики.

3. Модель Пуассона, использующая вероятностный подход для оценки количества голов, хорошо справлялась с прогнозированием вероятностей определённых счётов, однако при сведении задачи к классификации исходов (победа/ничья/поражение) давала более ограниченный результат из-за предположений о независимости событий и постоянстве среднего значения интенсивности.

В целом статистические методы подтвердили свою базовую полезность, но уступили более современным алгоритмам, особенно на сложных участках выборки.

Алгоритмы машинного обучения показали более высокие результаты по ключевым метрикам.

1. Деревья решений продемонстрировали адекватное качество прогнозов (accuracy около 70 %), но склонность к переобучению ограничила их практическую применимость.

2. Random Forest значительно улучшил ситуацию: ансамбль деревьев показал точность порядка 73–75 %, устойчиво работал с шумными данными и обеспечил более высокий уровень обобщения. Особенно полезным оказался механизм определения важности признаков, что позволило уточнить модель и улучшить её интерпретируемость.

3. Градиентный бустинг (в вариантах XGBoost и CatBoost) стал одним из наиболее эффективных алгоритмов. Его точность достигала 77–79 %, а F1-score показывал высокие значения для всех классов. Преимущество бустинга проявилось в способности выявлять тонкие нелинейные закономерности и эффективно работать с большим количеством зависимых признаков.

4. Нейросетевые модели продемонстрировали наиболее высокий уровень прогностической точности, особенно при наличии достаточного объёма данных для обучения. Полносвязная нейронная сеть, оптимизированная под структуру данных и дополненная механизмами регуляризации, показала точность в диапазоне 78–82 %.

Помимо абсолютных значений точности важно учитывать устойчивость прогнозов, анализ показал:

- статистические модели демонстрируют стабильность, но низкую адаптивность;
- модели машинного обучения устойчивы к шумам и дают высокие значения F1-score;
- нейронные сети показывают лучшую устойчивость при сложных взаимосвязях признаков, но требуют более тщательной подготовки данных.

Особое преимущество ансамблевых методов заключалось в минимизации ошибок: разброс значений по метрикам был значительно ниже, чем у отдельных моделей.

В совокупности именно ансамблевые алгоритмы и нейросетевые модели показали наиболее высокую эффективность и обоснованно могут рассматриваться как основа автоматизированных систем прогнозирования.

Разработка интеллектуальной системы прогнозирования спортивных событий явилась завершающим этапом исследования и позволила объединить отдельные аналитические компоненты в единую функциональную среду. Программный комплекс был спроектирован таким образом, чтобы обеспечить полный цикл работы с данными – от их загрузки и предобработки до построения прогноза и визуализации результатов. В отличие от традиционных статистических инструментов, ориентированных преимущественно на анализ отдельных показателей, интеллектуальная система реализует целостный подход, позволяющий пользователю работать с большими массивами данных и получать аналитически обоснованные предсказания в удобной интерактивной форме.

Заключение

Проведённое исследование позволило всесторонне рассмотреть современные подходы к прогнозированию спортивных событий и выявить наиболее эффективные методы анализа футбольных матчей. Особое внимание было уделено сравнительному изучению статистических моделей, алгоритмов машинного обучения и нейронных сетей, что позволило сформировать целостное представление о возможностях каждого класса методов и определить области их практического применения. Анализ показал, что традиционные статистические подходы сохраняют ценность благодаря своей интерпретируемости, однако в ряде случаев их предсказательная сила ограничивается линейностью используемых моделей и устойчивостью параметров.

Наиболее высокие результаты продемонстрировали алгоритмы машинного обучения – прежде всего ансамблевые методы градиентного бустинга и Random Forest, которые обладают способностью выявлять сложные зависимости и обеспечивать устойчивость при наличии шумов в данных. Применение нейронных сетей позволило дополнить картину: модели глубокого обучения продемонстрировали высокую точность при анализе нелинейных и слабо структурированных признаков, особенно в тех случаях, когда необходимо учитывать динамику формы команды и временные закономерности.

Практическим результатом исследования стало создание интеллектуальной программной системы, включающей инструменты обработки данных, построения моделей и визуализации результатов. Такая система обеспечивает удобный и интуитивно понятный механизм работы с массивами спортивной статистики и может использоваться как в учебных, так и в прикладных задачах. Возможность интеграции различных моделей, адаптивность архитектуры и реализованный механизм сравнения прогнозов делают разработанную систему гибким инструментом для анализа широкого спектра спортивных данных.

Таким образом, представленный в исследовании подход подтверждает высокую эффективность использования современных методов машинного обучения и нейросетевого моделирования при прогнозировании спортивных событий. Перспективы дальнейшего развития включают расширение набора данных, интеграцию текстовой аналитики для учёта новостного контекста, применение рекуррентных и трансформерных моделей для анализа временных последовательностей, а также создание гибридных систем, объединяющих несколько типов алгоритмов. Всё это будет способствовать повышению точности прогнозов и укреплению роли интеллектуальных технологий в области спортивной аналитики.

Список литературы

1. Андреев К. С. Компьютерное зрение и машинное обучение. СПб.: Питер, 2020.
2. Брук Т. Статистические методы в спортивной аналитике. М.: Спорт и наука, 2020.
3. Гудфеллоу Я., Бенджио И., Курвиль А. Глубокое обучение. М.: ДМК Пресс, 2018.
4. Половников Е. П. Технологии машинного обучения в автоматизированных системах. М.: Академия, 2021.
5. Bukreiev D. O. et al. Features of the use of software and hardware of the educational process in the conditions of blended learning // AET 2020-Symposium on Advances in Educational Technology. Technology (AET 2020). SCITEPRESS, 2022. №. 2. С. 236-244.
6. Bukreiev D. Neuro-network technologies as a mean for creating individualization conditions for students learning // SHS Web of Conferences. EDP Sciences, 2020. Т. 75. С. 04013.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ YOLOV8–YOLO11 В ЗАДАЧАХ ОБНАРУЖЕНИЯ ОБЪЕКТОВ В РЕАЛЬНОМ ВРЕМЕНИ

Корж А. А., Олейник Н. П.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: alex.korhz13@mail.ru,
nata.oleynik.2014@mail.ru,

Постановка проблемы. Отсутствие сопоставимых по методике сравнений YOLOv8–YOLO11 приводит к выбору моделей «по привычке» или по разрозненным публикациям, где различаются датасеты, размер входа и правила измерения задержки. Для практических систем это означает либо избыточные требования к вычислительным ресурсам, либо неприемлемое снижение качества детекции при переносе на целевую сцену.

Материалы и методы исследования

История семейства YOLO начинается с работы J. Redmon и соавт. (2015) [1], где предложен одноэтапный подход к детекции объектов. В последующих версиях развивались идеи ускорения и повышения точности; например, для YOLOv7 показано, что переработка backbone/neck и обучение с учётом задач реального времени позволяют сохранить высокую скорость при конкурентной точности [2].

Ultralytics в YOLOv8 перешли к anchor-free предсказаниям и унифицированному фреймворку задач, оставив NMS как обязательную часть конвейера [3].

В YOLOv9 введены GELAN и механизм PGI, рассчитанный на улучшение градиентного потока на этапе обучения [4]. Работа по YOLOv10 (Tsinghua University) предложила end-to-end детекцию без NMS через согласованное двойное назначение (one-to-many и one-to-one) и более лёгкие вычислительные блоки [5].

В YOLO11 Ultralytics сфокусировались на снижении параметрической сложности и уско-

рении CPU-инференса, заменив базовые блоки и расширив внимание (C2PSA) [6].

При этом в прикладной литературе преобладают оценки отдельных версий на COCO; сравнения по одинаковым метрикам задержки и потребления ресурсов для edge/CPU встречаются ограниченно.

Цель исследования – выполнить структурированный анализ YOLOv8–YOLO11, сопоставив архитектурные изменения и их влияние на mAP, задержку инференса и число параметров, и дать практические рекомендации выбора модели для типовых сценариев развёртывания.

Результаты исследования и их обсуждение

Все рассматриваемые версии (табл. 1) сохраняют трёхкомпонентную схему (backbone–neck–head), однако различаются тем, как организован поток признаков и как формируется обучающий сигнал. YOLOv8 использует блок C2f и SPPF, работает в anchor-free постановке и требует NMS. YOLOv9 вводит GELAN и PGI: вспомогательный путь для устойчивых градиентов используется на обучении и удаляется на инференсе, сохраняя классическую постобработку. YOLOv10 переносит часть сложности из постобработки в обучение: схема one-to-one позволяет выполнять вывод без NMS, что уменьшает задержку и упрощает конвейер. YOLO11 развивает линию повышения эффективности за счёт более компактных блоков (C3k2) и внимания (C2PSA), что важно для CPU и энергоограниченных систем.

По опубликованным бенчмаркам [4,5], различия точности между поколениями после YOLOv9 сравнительно невелики, тогда как различия по задержке и цене вывода заметнее. В частности, для малых конфигураций заявлено, что YOLOv10-S достигает латентности порядка 2,49 мс с учётом всей постобработки (без отдельного NMS), тогда как для YOLOv8-S приводится около 7,07 мс с учётом NMS. Для CPU-сценариев YOLO11 демонстрирует ускорение порядка 30%

для папо-модели, что критично при работе на видеосерверах без GPU и на edge-устройствах.

В прикладных областях выбор версии (табл. 2) определяется не только mAP, но и распределением задержки, памятью и устойчивостью. В автономном вождении важны стабильные 30 FPS и поведение при тумане, дожде и ночной сцене; отсутствие NMS в YOLOv10 снижает вариативность задержки, а модели YOLO11 среднего размера целесообразны, когда требуется более высокая точность при наличии GPU. В видеонаблюдении приоритетом становится непрерывная работа 24/7 и ограничение ложных тревог; компактные модели YOLO11n подходят для CPU-развёртывания и малой памяти, а YOLOv10s оправдана при наличии GPU и жёстком ограничении на задержку. В промышленном контроле качества критичны мелкие дефекты и высокая частота кадров; здесь полезны более точные конфигурации и, при необходимости, варианты с сегментацией, при этом ускорение вывода уменьшает нагрузку на линию инспекции.

Личный вклад авторов заключается в систематизации различий между версиями по единой схеме сравнения (архитектура, обучение, постобработка, ресурсы) и в формировании матрицы выбора для типовых сценариев. Представленная структура анализа может использоваться как шаблон для проверки новых выпусков YOLO, когда скорость обновления превышает скорость появления полноценных сравнительных публикаций.

Для прикладных внедрений требуется расширять набор проверок за пределы стандартных отчётов COCO: оценивать устойчивость к шуму, сжатию и артефактам видеопотока, проверять перенос на узкие доменные датасеты (аэрофото, подводная съёмка, медицинские изображения) с фиксированным бюджетом дообучения, а также измерять влияние INT8-квантования и дистилляции на качество и задержку на конкретном CPU/NPU.

Таблица 1

Сравнение ключевых компонентов YOLOv8–YOLO11

Компонент	YOLOv8	YOLOv9	YOLOv10	YOLO11
Backbone block	C2f	RepNCSPPELAN4 (GELAN)	C2f / C2fCIB (rank-guided)	C3k2
Downsampling	Conv (3×3, stride=2)	ADown (pooling-based)	SCDown (decoupled)	Conv (3×3, stride=2)
Neck attention	SPPF	SPPELAN	SPPF + PSA	SPPF + C2PSA
Необходимость NMS	Требуется	Требуется	Не требуется	Требуется (ускорена)
Эффективность параметров	Базовая	Выше (меньше параметров)	Высокая	Высокая
Особенности вывода	Стандартный конвейер	PGI только на обучении	End-to-end one-to-one	Ускорение CPU, DWConv

Рекомендации по выбору версии для типовых сценариев

Сценарий	Рекомендуемая версия	Основание выбора
Edge-устройства (Raspberry Pi, Jetson Nano)	YOLO11n	Малая модель, ускоренный CPU-инференс, низкое потребление памяти.
Видеонаблюдение 24/7	YOLO11n/s или YOLOv10s	YOLO11 – экономия ресурсов на CPU; YOLOv10 – минимальная задержка при наличии GPU и отсутствии NMS.
Автономное вождение	YOLOv10m или YOLO11m	YOLOv10 – более предсказуемая задержка без NMS; YOLO11m – выше точность при наличии вычислительного бюджета.
Промышленный контроль качества	YOLO11m (и варианты с сегментацией) / YOLOv10m	Нужны высокая точность по мелким объектам и высокая частота кадров; выбор зависит от доступного ускорителя.
Облачный сервис	YOLO11x или YOLOv10x	Приоритет точности; задержка на GPU приемлема, выбирается по нагрузке и стоимости вычислений.

Заключение

Отдельный интерес представляет сравнение не только средних значений, но и распределений латентности, поскольку именно предсказуемость задержки определяет пригодность детектора для реального времени.

Сопоставление YOLOv8–YOLO11 показывает, что эволюция после 2023 года в значительной мере направлена на снижение задержки и стоимости вывода при близких уровнях точности: YOLOv9 усиливает обучение через PGI, YOLOv10 устраняет необходимость NMS и тем самым уменьшает задержку, а YOLO11 сокращает число параметров и ускоряет CPU-инференс за счёт замены базовых блоков и расширения внимания, на практике это позволяет обоснованно выбирать версию под ограничения конкретного стенда (CPU/GPU, память, требуемый FPS) и сокращает риск неоправданных затрат при развёртывании, при этом дальнейшая проверка должна включать доменную переносимость, устойчивость к артефактам видеопотока и эффект квантования на целевом оборудовании.

Список литературы

1. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). P. 779-788. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf (date accessed: 18.12.2025).
2. YOLOv7: Trainable state-of-the-art object detector. arXiv preprint arXiv:2207.02696. URL: <https://arxiv.org/abs/2207.02696> (date accessed: 18.12.2025).
3. YOLOv8: A state-of-the-art real-time object detector. Ultralytics GitHub Repository. URL: <https://github.com/ultralytics/ultralytics> (date accessed: 15.12.2025).
4. YOLOv9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616. URL: <https://arxiv.org/abs/2402.13616> (date accessed: 20.12.2025).
5. YOLOv10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458. URL: <https://arxiv.org/abs/2405.14458> (date accessed: 18.12.2025).
6. YOLO11: Advanced Real-time Object Detection. Ultralytics Documentation. URL: <https://docs.ultralytics.com/> (date accessed: 16.12.2025).

ОБЗОР РАЗВИТИЯ И ТЕХНОЛОГИЧЕСКИХ ВЫЗОВОВ ПЛАТФОРМ ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО СПОРТИВНОМУ ПРОГРАММИРОВАНИЮ

Кремь А. А., Мозговенко А. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: alex.kremi@mail.ru, ya@amozgovenko.ru

Постановка проблемы. Рост массовых онлайн-соревнований по спортивному программированию, характеризующихся одновременной подачей тысяч решений, многоязычной поддержкой и разнообразием форматов задач, обнажил архитектурные ограничения существующих платформ автоматизированной проверки. Практика их эксплуатации показывает, что различия в способах изоляции исполнения кода, организации очередей компиляции и запуска, а также в механизмах масштабирования вычислительных ресурсов приводят к нестабильному времени проверки, трудновоспроизводимым результатам и рискам нарушения безопасности. Отсутствие систематизированного анализа архитектурных решений, подтверждённого сопоставлением реальных платформ и их инженерных компромиссов, затрудняет выработку обоснованных рекомендаций по проектированию устойчивых и детерминированных соревновательных систем, что формирует центральную проблему настоящего исследования.

Материалы и методы исследования

Анализ публикаций 2020–2025 годов демонстрирует, что исследовательский и инженерный фокус в области платформ для спортивного программирования сместился от описания базовой функциональности онлайн-проверки к решению прикладных задач масштабируемости, безопасной изоляции исполнения и воспроизводимости

результатов в условиях массовых конкурсов; это проявляется в появлении работ, систематизирующих типы автоматизированных систем оценки и выделяющих архитектурные узкие места при высоких нагрузках. Параллельно усилилось внимание к практическим инженерным решениям: исследования последних лет подробно рассматривают микросервисную декомпозицию, контейнеризацию и оркестрацию как основные приёмы обеспечения горизонтального масштабирования и быстрой реагируемости подсистем проверки, а также анализируют особенности балансировки очередей заданий и управления ресурсами в распределённых средах. Новое направление, заметно усилившееся в 2023–2025 годах, связано с оценкой семантической корректности кода и попытками использовать модели и инструменты на базе искусственного интеллекта для улучшения качества проверки – как дополнение к традиционным тест-наборным подходам, так и для создания гибридных фреймворков оценки, что ставит свои требования к архитектуре платформ и к процедурам валидации оценок. Практические отчёты и блоги команд крупнейших платформ подтверждают ускоренную эволюцию интерфейсов и инфраструктуры: разработчики отмечают рост объёмов трафика, расширение набора поддерживаемых языков и постоянные улучшения подсистем компиляции/изолированного исполнения в ответ на изменяющиеся потребности сообщества. Наконец, в ряде научных публикаций и сборников появляются систематизированные описания требований к online-judge-системам и их архитектурам, предлагающие формальные модели компонентов и критерии оценки проектных компромиссов, что создаёт основу для сопоставительного анализа и формализации рекомендаций по проектированию устойчивых соревновательных платформ.

Целью настоящего исследования является сравнительно-аналитическое осмысление архитектурных решений современных платформ для проведения соревнований по спортивному программированию с выявлением ключевых тенденций их развития и системных технологических вызовов, возникающих при масштабировании и усложнении соревновательной инфраструктуры.

Результаты исследования и их обсуждение

Анализ предметной области.

В качестве объектов анализа рассматривались типовые архитектурные компоненты таких систем, включая подсистемы приёма и хранения пользовательских решений, механизмы компиляции и исполнения программного кода, средства автоматизированной проверки результатов, а также сервисы управления соревнованиями и пользовательскими сессиями.

При отборе материалов учитывались платформы, обладающие устойчивой практикой эксплуатации, поддерживающие массовое участие и различные языки программирования, что позволило сосредоточиться на архитектурных решениях, доказавших свою работоспособность в условиях реальных соревновательных нагрузок.

В рамках исследования применялся структурно-функциональный анализ, позволивший сопоставить архитектурные элементы платформ с их функциональным назначением в соревновательном процессе. Такой подход дал возможность проследить, каким образом технические решения отражаются на ключевых характеристиках системы, включая пропускную способность, время отклика и детерминированность результатов проверки решений.

Анализ проводился с учётом типовых сценариев эксплуатации, характерных для соревнований различного уровня, от локальных учебных конкурсов до международных турниров с одновременным участием большого числа команд.

Для выявления эволюционных тенденций в архитектуре платформ использовался историко-технологический метод, основанный на сопоставлении ранних и современных архитектурных подходов. Рассматривались переходы от монолитных систем к модульным и распределённым архитектурам, а также изменение роли вычислительной инфраструктуры в процессе проверки решений. Это позволило проследить, каким образом рост вычислительных требований и усложнение форматов соревнований влияли на выбор архитектурных решений и стимулировали внедрение новых технологических подходов, таких как изоляция исполнения пользовательского кода и динамическое масштабирование ресурсов.

Дополнительно в исследовании применялся метод качественного анализа инженерных ограничений, возникающих при проектировании и эксплуатации платформ для спортивного программирования. Рассматривались типовые технологические вызовы, связанные с безопасностью исполнения недоверенного кода, обеспечением воспроизводимости результатов и поддержкой гетерогенных вычислительных сред. Анализ данных аспектов проводился без привязки к конкретным реализациям, что позволило обобщить результаты и сформулировать выводы, применимые к широкому классу соревновательных платформ.

В результате сравнительного анализа архитектурных решений современных платформ для проведения соревнований по спортивному программированию были рассмотрены ключевые представители данного класса цифровых систем, обладающие разной историей развития, функциональной направленностью и масштабом эксплуатации.

При выборе конкретных примеров в качестве предметов анализа ориентировались на платформы, которые активно используются сообществом программистов для проведения регулярных соревнований, обладают устойчивой инфраструктурой и представлены значительным числом участников. К числу таких платформ относятся Codeforces, CodeChef и HackerRank, каждая из которых иллюстрирует характерные архитектурные подходы и технологические вызовы, присущие современным соревновательным системам.

Codeforces – российская платформа для проведения онлайн-соревнований по спортивному программированию, созданной и поддерживаемой группой программистов во главе с Михаилом Мирзаяновым (ИТМО), широкая аудитория в более чем 600,000 зарегистрированных пользователей. Она функционирует как самостоятельная цифровая экосистема, где регулярно проводятся конкурсы, насчитываются рейтинги участников и обеспечивается автоматическая проверка решений. Платформа является одной из наиболее известных и крупных в мире соревновательного программирования, с миллионами зарегистрированных пользователей и активным сообществом программистов.

Анализ платформы позволяет выделить её как одну из наиболее массовых и технически зрелых систем в области соревновательного программирования. Платформа возникла как независимый проект и развивалась в направлении поддержки регулярных алгоритмических конкурсов с глобальной рейтинговой системой, что потребовало проектирования архитектуры, способной обрабатывать десятки тысяч одновременных запросов от участников в период проведения раундов. Codeforces реализует сложный механизм автоматической проверки решений, включающий приём исходного кода, компиляцию, исполнение в изолированной среде и сравнение результатов с тестовыми сценариями, что требует балансировки между скоростью выполнения и безопасностью, особенно в условиях мощных нагрузок во время пиковых периодов соревнований. Кроме того, наличие функционала для публикации задач, редакционных материалов и блогов участников создаёт дополнительные требования к интеграции компонентов управления контентом и соревновательной логики, что усложняет архитектуру системы и делает её мультисервисной по своей природе.

В противоположность Codeforces архитектура CodeChef отражает другой модельный подход, ориентированный на разнообразие форматов соревнований и длительность взаимодействия с пользователем. Платформа проводит как краткие конкурсы, так и более длительные соревнования, что формирует нагрузку на инфраструктуру в более разрозненные временные окна, но с большим объёмом задач и участников, стремящихся к разным типам участия.

Такое распределение нагрузок диктует необходимость гибкого механизма масштабирования ресурсов, в котором подсистемы проверки решений, очередей исполнения и базы данных должны динамически адаптироваться к изменяющемуся числу активных участников. Сама архитектура CodeChef ориентирована на обеспечение устойчивой производительности при вариативных условиях эксплуатации, что требует разработки продвинутых механизмов кеширования, оптимизации ввода-вывода и управления вычислительными ресурсами для минимизации времени отклика и предотвращения деградации качества обслуживания.

Платформа HackerRank представляет собой ещё один пример технологически ориентированной системы, но с более широким контекстом применения: помимо классических соревнований по алгоритмическому программированию, она активно используется для оценки профессиональных навыков разработчиков в контексте рекрутинга и технических собеседований.

Архитектурные решения HackerRank включают многоуровневую систему тестирования, охватывающую не только алгоритмы, но и практические задания на разнообразных языках программирования, что требует от инфраструктуры поддержки обширного набора компиляторов и сред выполнения. Такая мультиаспектная проверка решений создаёт дополнительные технологические вызовы, связанные с безопасным исполнением кода, детерминированностью результатов и консистентностью оценок при различной конфигурации вычислительных сред.

Подобные требования накладывают на архитектуру платформы обязательность строгой изоляции процессов, обеспечения репродуцируемости тестов и контроля за средой исполнения, что в совокупности увеличивает сложность реализации и эксплуатацию данных систем.

Выводы

Благодаря анализу архитектурных подходов к созданию платформ для проведения соревнований по спортивному программированию можно сделать ряд обобщённых выводов о текущем состоянии и ключевых технологических вызовах этой области.

Эволюция систем от монолитных реализаций к модульным и распределённым архитектурам продемонстрировала, что центральными свойствами успешной платформы являются способность обеспечивать надёжную изоляцию выполнения пользовательского кода, детерминированность результатов проверки и гибкость в управлении вычислительными ресурсами; именно эти требования определяют выбор инструментов контейнеризации, песочниц, очередей заданий и механизмов масштабирования. Проведенный анализ архитектурных особенностей указанных платформ выявил несколько

общих технологических вызовов, характерных для подобных цифровых экосистем.

Во-первых, это обеспечение изоляции выполнения кода: платформы должны гарантировать, что решения участников не вмешиваются в работу основной системы и не нарушают её целостность, что требует применения контейнеризации, песочниц и строгих политик безопасности.

Во-вторых, необходима детерминированность условий проверки: одинаковый алгоритм должен получить аналогичный результат при любых условиях запуска, что требует стандартизованных сред выполнения и чётко определённых ограничений по времени и памяти.

В-третьих, важнейшим вызовом является масштабируемость архитектуры: системы обязаны обеспечивать надёжную работу при пиковых нагрузках, что предполагает автоматическое распределение нагрузки между вычислительными узлами, оптимизацию очередей заданий и грамотное управление ресурсами [1].

Наконец, интеграция аналитических и социальных функций, таких как рейтинги, форумы и образовательные материалы, требует от архитектуры гибкости и расширяемости, поскольку подобные компоненты должны не только работать автономно, но и поддерживать consistente взаимодействие с основными модулями системы проверки.

Практика показала, что неправильная композиция системы на сервисы или пренебрежение стандартами среды выполнения приводит к трудноуловимым ошибкам воспроизводимости и к рискам безопасности, тогда как продуманная модульность упрощает эволюцию функционала и снижает стоимость сопровождения.

Масштабируемость остаётся критическим фактором – как в аспекте обработки пиковых нагрузок во время массовых контестов, так и в обеспечении низкой задержки отклика для интерактивных задач – и требует внедрения адаптивных стратегий балансировки, продвинутой политики очередей и мониторинга загрузки. Кроме того, мультиформатность современных соревнований и потребность в поддержке множества языков программирования ставят дополнительные требования к унификации интерфейсов компиляции и исполнения, к стандартизации ограничений по времени и памяти, а также к обеспечению репродуцируемости тестовых сред.

Наконец, интеграция социальных, рейтинговых и образовательных компонентов в соревновательную инфраструктуру предъявляет высокие требования к согласованности данных и гибкости взаимодействия подсистем: платформа должна не только точно оценивать корректность решений, но и поддерживать прозрачную обратную связь, аналитические отчёты и инструменты для обучения участников.

Исходя из этих наблюдений, целесообразно направлять дальнейшие инженерные усилия на развитие модульных архитектур с явной границей ответственности сервисов, стандартизацию сред выполнения и тестовых спецификаций, внедрение механизмов автоматического масштабирования и продвинутых средств наблюдения за поведением системы; параллельно необходимы исследования по формализации критериев детерминированности и воспроизводимости в контексте соревновательной оценки, а также по разработке методик экспериментального сравнения архитектурных решений в реальных нагрузочных сценариях.

Такое сочетание практических улучшений и методологического осмысления будет способствовать созданию более надёжных, безопасных и педагогически ценных платформ, способных удовлетворять растущие требования сообществ программистов и образовательных учреждений.

Список литературы

1. Евстропов Г. О. Системы оценивания в задачах с автоматической проверкой решений // Информационный журнал по олимпиадам и программированию. Информатика и образование. 2016. № 3. С. 65-67.
2. Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г. Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 61-73.
3. Корнеев Г. А., Елизаров Р. А. Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. 2003. № 1. С. 61-73.
4. Материалы VIII Всероссийской с международным участием научно-практической конференции «День спортивной информатики», (Москва, 2 декабря 2024 года) / Ассоциация компьютерных наук в спорте. Москва, 2024. 132 с.

ПУЗЫРЬКОВАЯ СОРТИРОВКА: СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМА НА СОВРЕМЕННЫХ ЯЗЫКАХ ПРОГРАММИРОВАНИЯ В ЧИТАЕМОСТИ КОДА

Кузнецова В. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: skald.beregovoi@yandex.ru*

Научный руководитель: Береговой А. В.

Введение

Пузырьковая сортировка (Bubble Sort) – один из старейших алгоритмов сортировки, известный еще с середины XX века. Его название, впервые введенное в работе Айверсона (Iverson) в 1962 году, точно отражает суть процесса: элементы с наибольшими (или наименьшими) значениями постепенно «всплывают» к своему законному месту в конце (или начале) массива, подобно пузырькам воздуха в воде. [1]

Цель исследования – на основе различных современных языков программирования (C++,

Java, JavaScript, Go и Python) произвести сравнительный анализ представлений визуальное представление пузырькового метода сортировки.

Материалы и методы исследования

В статье исследуется алгоритм, его способы оптимизации и реализация на различных языках программирования: C++, Java, JavaScript, Go и Python, сравнение разных параметров кода, в особенности его читаемость и простота понимания.

Результаты исследования и их обсуждение

Базовый принцип работы: Алгоритм многократно проходит по массиву, попарно сравнивая соседние элементы. Если два со-

седа расположены в неверном порядке относительно выбранного критерия (например, по возрастанию), алгоритм меняет их места. Этот процесс повторяется до тех пор, пока весь массив не будет упорядочен, то есть пока не будет выполнен полный проход без единой перестановки. В обширном арсенале алгоритмов сортировки, где доминируют высокоэффективные методы вроде быстрой сортировки (QuickSort), сортировки слиянием (MergeSort) и пирамидальной сортировки (HeapSort), пузырьковая сортировка занимает особую, почти архаичную нишу.

Рассмотрим детальную пошаговую работу алгоритма на конкретном примере. Возьмем массив целых чисел, который требуется отсортировать по возрастанию: `[5, 3, 8, 4, 2]`.

Первый проход ($i = 0$):

- Сравнение индексов 0 и 1: `5 > 3` → истина → меняем местами. Массив: `[3, 5, 8, 4, 2]`
- Сравнение индексов 1 и 2: `5 > 8` → ложь → массив без изменений: `[3, 5, 8, 4, 2]`
- Сравнение индексов 2 и 3: `8 > 4` → истина → меняем местами. Массив: `[3, 5, 4, 8, 2]`
- Сравнение индексов 3 и 4: `8 > 2` → истина → меняем местами. Массив: `[3, 5, 4, 2, 8]`

Результат первого прохода: Наибольший элемент (8) «всплыл» на свою конечную позицию. Неотсортированная часть массива теперь `[3, 5, 4, 2]`.

Второй проход ($i = 1$):

- Сравнение индексов 0 и 1: `3 > 5` → ложь → массив без изменений: `[3, 5, 4, 2, 8]`
- Сравнение индексов 1 и 2: `5 > 4` → истина → меняем местами. Массив: `[3, 4, 5, 2, 8]`
- Сравнение индексов 2 и 3: `5 > 2` → истина → меняем местами. Массив: `[3, 4, 2, 5, 8]`

Результат второго прохода: Второй по величине элемент (5) занял свою позицию. Неотсортированная часть: `[3, 4, 2]`.

Третий проход ($i = 2$):

- Сравнение индексов 0 и 1: `3 > 4` → ложь → массив без изменений.
- Сравнение индексов 1 и 2: `4 > 2` → истина → меняем местами. Массив: `[3, 2, 4, 5, 8]`

Результат третьего прохода: Элемент 4 занял свою позицию. Неотсортированная часть: `[3, 2]`.

Четвертый проход ($i = 3$):

- Сравнение индексов 0 и 1: `3 > 2` → истина → меняем местами. Массив: `[2, 3, 4, 5, 8]`

Результат четвертого прохода: Массив полностью отсортирован. Алгоритм завершает работу.

Базовая версия алгоритма неэффективна, так как продолжает работу даже после того, как массив стал отсортированным. Существует две ключевые оптимизации:

1. Флаг обменов (Swapped Flag): На каждом проходе отслеживается, был ли совершен хотя бы один обмен. Если внутренний цикл завершился без единой перестановки, это означает, что массив уже отсортирован, и внешний цикл можно прервать досрочно. Именно эту версию мы будем реализовывать в данной статье.

2. Учет последней перестановки (Last Swap Index): Более продвинутая оптимизация. Вместо флага запоминается индекс последней выполненной перестановки во внутреннем цикле. На следующей итерации внутренний цикл

будет выполняться только до этого индекса, а не до фиксированного $n - i - 1$. [3]

Теперь, имея теоретическое понимание алгоритма, перейдем к практической части. Мы реализуем оптимизированную версию с флагом обменов на пяти различных языках. Наша цель – не сравнение производительности (где компилируемые языки, такие как C++ и Go, заведомо выигрывают у интерпретируемого Python), а анализ читаемости, лаконичности кода.

```

Реализация на C++[6]
#include<iostream>
#include<vector>
using namespace std;
int main(){
vector<int> v={64,34,25,12,22,11,90};
for(int i=0,n=v.size();i<n-1;i++){

```

```
bool s=0;
for(int j=0;j<n-i-1;j++)
    if(v[j]>v[j+1]) swap(v[j],v[j+1]),s=1;
if(!s) break;}
cout<<«Отсортированный массив: «;
for(int n:v) cout<<n<<« »;
return 0;}
```

Анализ кода на C++:

– Управление памятью и типами: Явное указание типов обеспечивает безопасность и предсказуемость, но добавляет «синтаксического шума».

– Процесс обмена элементов: Классический трехшаговый обмен с временной переменной `temp` является универсальным, но выглядит громоздко и не является идиоматическим для современного C++.

– Читаемость: Код достаточно читаем для опытного разработчика, но может показаться перегруженным новичку из-за необходимости разбираться с заголовочными файлами, областью видимости (`std::`) и ручным управлением итерациями.

Реализация на Java

```
public class B {public static void main(String[]a) {
int[]n={64,34,25,12,22,11,90},t=n.clone();
for(int i=0,l=t.length;i<l-1;i++){
boolean s=false;
for(int j=0;j<l-i-1;j++)
if(t[j]>t[j+1]){int x=t[j];t[j]=t[j+1];t[j+1]=x;s=true;}
if(!s)break;}
System.out.print(«Отсортированный массив: «);
for(int x:t)System.out.print(x+« »);}}
```

Анализ кода на Java:[4]

– Объектно-ориентированный каркас: Необходимость помещать код в класс `BubbleSort` и использовать модификатор `static` для простой процедурной функции добавляет церемониальности, которая не связана напрямую с логикой алгоритма.

– Синтаксическая схожесть с C++: Java унаследовала много синтаксических конструкций от C/C++, включая объявление переменных, циклы `for` и условные операторы `if`. Это делает код очень похожим на C++ версию со всеми ее достоинствами и недостатками.

– Читаемость: Для разработчиков, знакомых с C-подобными языками, код интуитивно понятен.

Реализация на JavaScript [7]

```
function bubbleSort(arr) {
let n = arr.length;
let swapped;
for (let i = 0; i < n - 1; i++) {
swapped = false;
for (let j = 0; j < n - i - 1; j++) {
if (arr[j] > arr[j + 1]) {
```

```
[arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];
swapped = true;}}
if (!swapped) {
break;}}
return arr;}
const numbers = [64, 34, 25, 12, 22, 11, 90];
console.log(«Исходный массив:», numbers);
console.log(«Отсортированный массив:»,
bubbleSort(numbers));
```

Анализ кода на JavaScript:

– Динамическая типизация: Отсутствие объявлений типов делает код менее многословным, но может скрывать потенциальные ошибки, которые в статически типизированных языках были бы выявлены на этапе компиляции.

– Деструктурирующее присваивание: Это мощная особенность современного JavaScript (ES6+), которая кардинально меняет читаемость критической части алгоритма – обмена элементов. Конструкция `[a, b] = [b, a]` интуитивно понятна и выразительна.

– Функциональный стиль: Возврат массива из функции (`return arr`) позволяет использовать вызов в выражениях, например, сразу в `console.log`, что удобно.

– Читаемость: Благодаря деструктуризации и лаконичному синтаксису, код на JavaScript становится одним из самых читаемых среди C-подобных языков.

Реализация на Go[5]

```
package main
import «fmt»
func bubbleSort(arr []int) {
n := len(arr)
var swapped bool
for i := 0; i < n-1; i++ {
swapped = false
for j := 0; j < n-i-1; j++ {
if arr[j] > arr[j+1] {
arr[j], arr[j+1] = arr[j+1], arr[j]
swapped = true}}
if !swapped {
break}}}}
func main() {
numbers := []int{64, 34, 25, 12, 22, 11, 90}
fmt.Println(«Исходный массив:», numbers)
bubbleSort(numbers)
fmt.Println(«Отсортированный массив:», numbers)}
```

Анализ кода на Go:

– Простота и минимализм: Синтаксис Go намеренно прост и лишен многих особенностей других языков (нет классов, исключений в классическом понимании, перегрузки операторов). Это делает код последовательным и предсказуемым.

– Статическая типизация с лаконичностью: Как и в C++/Java, типы указываются явно (`[]int`),

что обеспечивает безопасность. Однако краткая форма объявления переменных (`n := len(arr)`) и автоматический вывод нулевых значений делают код менее многословным.

– Параллельное присваивание: Способ обмена элементов `a, b = b, a` является в Go идиоматическим и столь же элегантным, как и в Python.

– Читаемость: Код на Go исключительно читаем. Отсутствие лишних скобок и ключевых слов, а также встроенный обмен значениями ставят Go очень высоко в рейтинге читаемости.

Python

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n - 1):
        swapped = False
        range(n - i - 1)
        for j in range(n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
        if not swapped:
            break
    return arr
```

Анализ и итоговое сравнение

1. Синтаксис как псевдокод (Максимальная читаемость):

– Код на Python практически неотличим от формального псевдокода, используемого для описания алгоритмов в учебниках. Ключевые слова `def`, `for...in range`, `if not` читаются как предложения на английском языке.

– Сравнение: C++ и Java требуют объявления типов и управления областью видимости с помощью фигурных скобок `{}`. Python использует обязательные отступы, которые одновременно являются и синтаксической конструкцией, и гарантией хорошо отформатированного, «красивого» кода.

2. Элегантность в критических операциях (Параллельное присваивание):

– Конструкция `arr[j], arr[j+1] = arr[j+1], arr[j]` является идиоматической для Python и решает одну из самых частых задач в алгоритмах – обмен значений – с беспрецедентной ясностью. Она интуитивно понятна даже тем, кто видит ее впервые.

– Сравнение: C++ и Java вынуждены использовать трехшаговый обмен с временной переменной, который является более низкоуровневым и заставляет мозг тратить дополнительные усилия на его «разбор». JavaScript и Go имеют схожие элегантные конструкции, что ставит их в один ряд с Python по этому параметру.

3. Выразительность и лаконичность:

– Объявление списка `numbers = [64, 34, ...]` предельно просто.

– Создание копии для чистоты эксперимента `copy()` – это один ясный метод.

– Сравнение: Python делает максимум с минимумом синтаксических элементов. В нем почти нет «церемониального» кода, не связанного напрямую с решаемой задачей.

4. Динамическая типизация и полиморфизм:

– Хотя динамическая типизация и может быть источником ошибок, в контексте учебных алгоритмов она является благом. Одна и та же функция `bubble_sort` может работать с любыми типами данных, для которых определен оператор сравнения `>` (числа, строки и т.д.), без необходимости писать шаблоны или перегружать функции.

– Сравнение: В C++ и Java для этого потребовалось бы использовать шаблоны (C++) или дженерики (Java) или перегружать метод для разных типов, что значительно усложнило бы учебный пример.

5. Баланс между абстракцией и контролем:

– Python, будучи языком высокого уровня, эффективно абстрагируется от низкоуровневых деталей (управление памятью, явные типы переменных в теле функции), позволяя программисту сосредоточиться на сути алгоритма – его логике. Это идеально для образовательных целей.

Заключение

Проведенный анализ однозначно показывает, что приоритеты в программировании могут быть разными. Если конечной целью является максимальная производительность вычислений для сортировки миллионов записей, то компилируемые языки, такие как C++ и Go, будут вне конкуренции.

Однако если мы говорим о педагогической ценности, скорости прототипирования, сопровождаемости кода и, что самое главное, о чистой читаемости и выразительности, то Python демонстрирует безоговорочно лучший результат. Пузырьковая сортировка, будучи простым алгоритмом, служит идеальным полигоном для демонстрации этой философской разницы. Python-реализация превращает ее из набора инструкций для машины в ясное и лаконичное описание алгоритма для человека.

Преимущество Python заключается в его способности минимизировать когнитивную нагрузку на программиста, позволяя ему сосредоточиться на решении задачи, а не на преодолении синтаксических барьеров языка. И в этой дисциплине он был и остается эталоном, на фоне которого даже такие изящные и современные языки, как Go и JavaScript, хоть и приближаются к нему, но все же уступают в чистоте и ясности изложения мысли.

Список литературы

1. Колтыгин Д. С., Баева А. Ю. Сравнительный анализ быстродействия методов сортировки массивов URL: <https://www.elibrary.ru/item.asp?id=43030420> (дата обращения: 15.12.2025).

2. Осипов А. А. Сравнение скорости сортировки (Bubble, Quick, Merge, Heap) // Перспективы развития науки и мирового сообщества: научно-методические и практические аспекты: материалы IX Международной научно-практической конференции. Физико-математические науки. 2024. URL: <https://innova-science.ru/wp-content/uploads/2025/06/sbornik-nauchnyh-trudov-19.06.2025-prs-9.pdf#page=54> (дата обращения: 15.12.2025).

3. Валиева Э. Р., Лябах К. Ю., Мазяков А. В. Методы сортировок и их эффективность // Актуальные проблемы интеграции науки и образования в регионе: материалы Всероссийской научно-практической конференции (с международным участием) (г. Бузулук, 20–21 мая 2021 года). Бузулук: Оренбург. гос. ун-т, 2021. С. 227–231.

4. Реализация пузырьковой сортировки на Java // Javarush. URL: <https://javarush.com/groups/posts/634-realizacija-puzirihkovoju-sortirovki-na-java> (дата обращения: 15.12.2025).

5. Сортировка пузырьком (Go) // Expanse.ru. URL: [https://expanse.ru/e/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC_\(Go\)](https://expanse.ru/e/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC_(Go)) (дата обращения: 15.12.2025).

6. Bubble Sort in C++ // Code of Code. URL: <https://codeofcode.org/lessons/bubble-sort-in-cpp/> (дата обращения: 15.12.2025).

7. Bubble Sort: изучаем самый простой алгоритм сортировки // Proglib.io. URL: <https://proglib.io/p/bubble-sort> (дата обращения: 15.12.2025).

КОНФИДЕНЦИАЛЬНОСТЬ ПЕРСОНАЛЬНЫХ ДАННЫХ В СОЦИАЛЬНЫХ СЕТЯХ: РИСКИ И СПОСОБЫ ИХ МИНИМИЗАЦИИ

Кузьменко Р. Д.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: anna.dyachenko597@mail.ru

Научный руководитель: Дяченко А. С.

Введение

Социальные сети в современном обществе занимают центральное место в системе коммуникации, распространения информации и организации повседневной деятельности пользователей [1]. Их активное развитие сопровождается ростом объема обрабатываемых персональных данных, что существенно повышает риски нарушения конфиденциальности [2]. В условиях цифровизации утечки информации, взломы пользовательских аккаунтов и несанкционированный доступ к персональным данным становятся одной из наиболее значимых угроз информационной безопасности [3].

Актуальность проблемы обусловлена ростом числа пользователей и усилением последствий киберинцидентов, что требует комплексного анализа рисков и разработки эффективных мер их минимизации.

Материалы и методы исследования

Анализ научных публикаций показывает, что проблема защиты персональных данных в социальных сетях рассматривается в условиях цифровой трансформации и роста объемов обрабатываемой информации. Исследователи отмечают, что социальные платформы являются приоритетной целью киберпреступности из-за концентрации персональных данных и высокой активности пользователей [2]. Современные работы подчёркивают многофакторный характер рисков конфиденциальности, при котором технические уязвимости усиливаются поведенческими и организационными факторами, что обосновывает необходимость комплексного подхода к защите персональных данных.

Целью данного исследования является анализ рисков нарушения конфиденциальности персональных данных пользователей социальных сетей, их классификация, а также определение наиболее эффективных способов минимизации выявленных угроз.

Результаты исследования и их обсуждение

В ходе исследования были проанализированы статистические данные об утечках персональных данных, а также обобщены типовые риски, характерные для социальных сетей [3, 4]. Полученные результаты позволили не только выявить масштабы проблемы, но и систематизировать основные виды угроз конфиденциальности.

Для количественной оценки динамики утечек персональных данных была проведена сравнительная характеристика показателей за 2024–2025 годы (таблица 1) [4].

Анализ статистики показывает, что при незначительном снижении числа инцидентов наблюдается резкое увеличение объема утёкших данных, что свидетельствует об укрупнении масштабов отдельных утечек и росте потенциального ущерба для пользователей [4].

Таблица 1

Динамика утечек персональных данных в России в 2024–2025 гг.

Показатель	2024 год	2025 год
Количество зарегистрированных инцидентов утечек	310	281
Объём утёкших персональных данных (млрд строк)	3,5	≈ 13
Темп изменения объёма утечек	—	рост в 3,7 раза
Основные цифровые источники утечек	Социальные сети, онлайн-сервисы	Социальные сети, мессенджеры

Классификация рисков нарушения конфиденциальности в социальных сетях

Категория рисков	Примеры угроз
Технические	Уязвимости программного обеспечения, фишинговые атаки, вредоносное ПО (malware), сторонние приложения
Поведенческие	Чрезмерное раскрытие личной информации, открытые профили, использование слабых паролей
Организационные	Внутренние утечки данных на платформах, недостаточный уровень защиты информации

В рамках исследования также была проведена классификация рисков нарушения конфиденциальности персональных данных в социальных сетях. Для систематизации выявленных угроз они были разделены на три основные категории (таблица 2).

Проведённая классификация показывает, что риски нарушения конфиденциальности носят взаимосвязанный характер. Технические уязвимости цифровых платформ существенно усиливаются поведенческими факторами, такими как неосторожное обращение пользователей с личной информацией, что в совокупности может приводить к массовым утечкам данных [2].

Полученные результаты также позволили определить основные направления минимизации рисков нарушения конфиденциальности персональных данных. Эффективная защита информации требует комплексного подхода, включающего индивидуальные, технические, организационные и правовые меры [5].

1. На индивидуальном уровне ключевыми мерами являются использование двухфакторной аутентификации, настройка параметров приватности профиля, регулярная смена сложных паролей, а также повышение уровня цифровой грамотности пользователей. Данные меры позволяют существенно снизить вероятность несанкционированного доступа к аккаунтам.

2. Технические меры направлены на защиту каналов передачи данных и устройств пользователей. К ним относятся использование VPN и технологий шифрования трафика, антивирусных программ, а также применение приложений с end-to-end шифрованием, обеспечивающим защиту информации на всём пути её передачи.

3. Организационные меры предполагают повышение ответственности самих цифровых платформ. В их число входят внедрение систем искусственного интеллекта для выявления угроз, локализация персональных данных на серверах Российской Федерации и проведение регулярных аудитов информационной безопасности [6, 7].

4. Правовые меры ориентированы на совершенствование нормативно-правовой базы, включая усиление законодательства в сфере защиты персональных данных и расширение тре-

бований Федерального закона № 152-ФЗ с учётом использования технологий искусственного интеллекта при обработке информации [5].

Комплексное применение указанных мер способно снизить риски утечек персональных данных на 70–80 %, однако требует постоянного мониторинга и обновления механизмов защиты.

Заключение

Проведённое исследование показало, что риски нарушения конфиденциальности персональных данных в социальных сетях продолжают возрастать, а характер киберугроз становится более сложным. Рост объёма утечек данных усиливает негативные последствия для пользователей. Классификация рисков и анализ мер защиты подтверждают необходимость комплексного подхода, включающего технические, организационные и правовые меры, а также повышение цифровой грамотности пользователей.

Список литературы

1. Дяченко А. С., Мухина Н. Ю. Цифровая этика в эпоху данных: баланс приватности, безопасности и свободы в обществе и образовании // Философские, социологические и психолого-педагогические проблемы современного образования. 2025. № 7. С. 42–47. EDN: NANHTA.
2. Social Media Privacy: Definition and Risks // Kaspersky Resource Center [Электронный ресурс]. URL: <https://www.kaspersky.ru/resource-center/definitions/social-media-privacy> (дата обращения: 23.12.2025).
3. Personal Data Leak Check // CyberNews [Электронный ресурс]. URL: <https://cybernews.com/personal-data-leak-check/> (дата обращения: 30.12.2025).
4. Вот сколько персональных данных россиян слили в Сеть с начала 2025 года // РБК Life [Электронный ресурс]. URL: <https://www.rbc.ru/life/news/68d26aaf9a794747a147fe5c> (дата обращения: 15.12.2025).
5. О персональных данных: Федеральный закон РФ от 27.07.2006 г. № 152-ФЗ (ред. действующая) [Электронный ресурс]. // КонсультантПлюс. URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 18.12.2025).
6. Тимофеев Б. И. Роль информационных технологий в управлении устойчивым развитием // Материалы XXV Всероссийской научно-практической конференции молодых ученых, аспирантов и студентов с международным участием, посвященной 80-летию Победы в Великой Отечественной войне 1941–1945 гг. (Нерюнгри, 23–25 октября 2025 г.). Якутск: Северо-Восточный федеральный университет им. М. К. Аммосова, 2025. С. 455–460. EDN: GGMJXB.
7. Шерман В. А. Анализ существующих систем управления содержанием // Студенческий научный форум: материалы XVIII международной студенческой научной конференции (Москва, 20 октября 2024 г. – 15 февраля 2025 г.). Москва: ООО «Евразийская научно-промышленная палата», 2025. С. 117–119. EDN: MOGAVJ.

КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА ВЕБ-ИНТЕРФЕЙСА УПРАВЛЕНИЯ УМНЫМ ДОМОМ

Кузьменко И. А., Мозговенко А.А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@amozgovenko.ru

Существующие решения для управления умным домом требуют создания эффективной и масштабируемой архитектуры, обеспечивающей безопасное взаимодействие между компонентами системы в режиме реального времени.

Материалы и методы исследования

Современные исследования в области автоматизации жилья акцентируют внимание на важности интеграции различных устройств в единую систему. Особое значение придается обеспечению безопасности передачи данных и возможности масштабирования системы.

Целью работы является разработка и анализ клиент-серверной архитектуры веб-интерфейса управления умным домом, обеспечивающей:

- Надежное взаимодействие компонентов системы
- Безопасную передачу данных
- Возможность масштабирования
- Работу в режиме реального времени

Результаты исследования и их обсуждение

Серверная часть реализована на языке программирования Node.js и современного фреймворка Express. Для правильной настройки сервера будут использоваться следующие протоколы:

- протокол HTTPS
- протокол MQTT
- WebSockets

Протокол HTTPS распределяется на две части:

- HTTP(HyperText Transfer Protocol) – относится к списку не защищенных протоколов
- HTTPS(HyperText Transfer Protocol Secure) – протокол является расширением протокола HTTP с поддержкой шифрования для повышения уровня безопасности

Согласно схеме взаимодействия сервера с приложением (рис. 1) все участники одной сети контактируют между собой с помощью сервера. Чтобы обеспечить непрерывную передачу данных между сервером и веб-приложением, используется подключение по WebSocket-у, и по протоколу подключения MQTT отправляются данные с умных устройств на сервер.

Серверная часть разработана по стандарту CRUD:

1. Create – создание данных
2. Read – считывание данных
3. Update – обновление данных
4. Delete – удаление данных

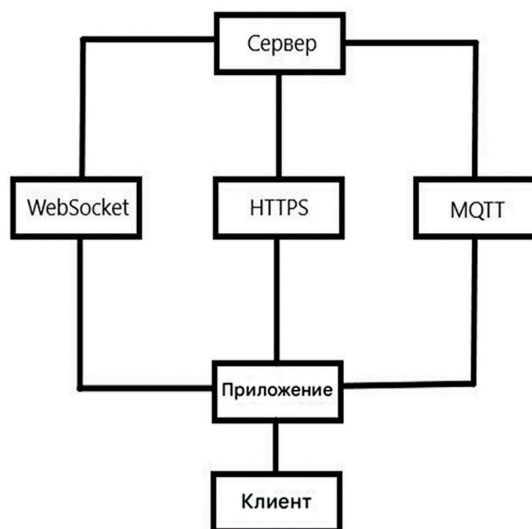


Рис. 1. Схема взаимодействия сервера с приложением

Хранилищем для хранения серверных данных выступает MongoDB. MongoDB представляет собой документно ориентированную базу данных и относится к типу NoSQL баз данных. MongoDB не является реляционной базой данных и использует коллекции и документы. Основной единицей данных служат документы, состоящие из пар ключ-значений. В коллекциях содержатся наборы функций и документов, являющихся эквивалентом таблиц реляционной базы данных.

MongoDB широко используется в сотрудничестве с AWS, Azure и другими источниками данных для разработки и функционирования приложений. Позволяя хранить и запрашивать большие объемы данных, он предлагает следующие надежные функции:

- Лучшее выполнение запросов с надлежащими функциями индексации и обработки.
- Аналитика в реальном времени и оптимизированная обработка данных с использованием специальных запросов.
- Улучшена доступность и гибкость данных благодаря надежным функциям репликации.
- Совместное использование данных позволяет разделять большие блоки данных для распределенного и более быстрого процесса выполнения запросов.

MongoDB хранит данные в формате JSON с парами ключей и значений для каждой сущности (рис. 2), в то время как базы данных SQL хранят данные в виде записи в строке таблицы (рис. 3).

MongoDB является лучшей базой для горизонтального и простого масштабирования. Гибкая база данных, которую можно постоянно совершенствовать, добавлять больше серверов, расширять хранилища и настраивать их и имеет

такие преимущества, как высокая производительность, простота развертывания, простота использования и удобное хранение данных. Данные группируются и хранятся в наборе данных. Каждый набор данных представляет собой набор, каждая база данных содержит несколько наборов.

На рисунке 4 наглядно представлена клиентская часть, на которой реализована форма входа и форма регистрации для новых пользователей.

Веб-приложение разработано на языке программирования TypeScript с использованием одной из самых популярных библиотек React. TypeScript является тем же языком программирования JavaScript, но имеет более расширенные возможности, чем JavaScript и более типизированную структуру кода.

Для обеспечения соединения сервера с приложением используются FETCH запросы и для получения данных в реальном времени используется WebSocket.

```

1  {
2    "orderId": 12345,
3    "shopperName": "Ivan Ivanov",
4    "shopperEmail": "ivanov@example.com",
5    "contents": [
6      {
7        "productID": 34,
8        "productName": "Super product",
9        "quantity": 1
10     },
11     {
12       "productID": 56,
13       "productName": "Wonderful product",
14       "quantity": 3
15     }
16   ],
17   "orderCompleted": true
18 }

```

Рис. 2. Формат данных JSON

Name	age	contact-mobile	home-address
Perry	20	9273723723	perry street

Рис. 3. Формат данных SQL

Smart Home System

Email placeholder

Password placeholder

[Login](#)

[Registration](#)

Рис. 4. Smart Home Systems форма входа

FETCH запросы делятся на методы:

- POST – для отправки данных на сервер
- GET – для получения данных с сервера
- PATCH – для обновления данных на сервере
- DELETE – для удаления данных с сервера

Чтобы открыть новый веб-сокет соединения, сначала нужно сделать запрос на подключение указав ссылку со специальным протоколом ws, равно как и HTTP, веб-сокет может быть связан с зашифрованным протоколом wss (рис. 5). Следует отметить, что лучше использовать зашифрованные протоколы подключения к серверам, так как именно такие протоколы гарантируют безопасный транспортный уровень и шифрует все данные от отправителя и расшифровывает их на стороне защитника от злоумышленников.

После инициализации нового сокета подключения следует разработать общение между сервером и приложением (рис. 6).

Переменная socket представляет собой объект с четырьмя функциями:

- onopen – функция, устанавливающая подключение между сервером и приложением
- onmessage – функция, которая после успешно установленного соединения передает данные с сервера в приложение.
- onclose – функция, закрывающая подключение между сервером и приложением

В случае разорванного соединения или ошибки сервера

- onerror – функция, выводящая все ошибки сервера

Для хранения серверных данных используется state-management библиотека Redux с использованием библиотеки redux-thunk для обработки асинхронных запросов. Библиотека Redux включает в себя основные модули для разработки:

- Slice(Reducer)
- Actions
- Selector's

Самым главным модулем является Slice, в этом модуле разрабатываются редьюсеры и сохраняется логика обработки и записи данных в переменную через actions. Вслед за Slice по приоритетности следуют Actions, где хранятся асинхронные функции отправляющие и получающие данные с сервера, сами actions являются очень гибкими, в них можно описывать не только логику обработки серверных данных, но и управлять состоянием компонента на «расстоянии», передавать локальные данные, уход. Selector's представляют собой функции в которых описаны ссылки на переменные в Slice, иными словами, через Selectors извлекаются серверные данные.

```
const socket = new WebSocket('wss://javascript.info');
```

Рис. 5. Веб-сокет с зашифрованным протоколом

```
socket.onopen = function (event) {
  console.log('[open] Connected', event);
  console.log('Sending data tto the server');
};

socket.onmessage = function (event) {
  console.log(`[message] Getting data from the server: ${event.data}`);
};

socket.onclose = function (event) {
  if (event.wasClean) {
    console.log(`[close] Connection closed, code=${event.code} reason=${event.reason}`);
  } else {
    console.log('[close] Connection dropped');
  }
};

socket.onerror = function (error) {
  console.log(`[error] ${error.message}`);
};
```

Рис. 6. Общение между сервером и приложением

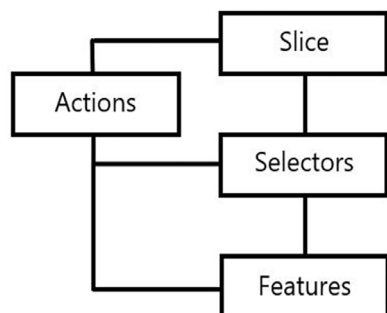


Рис. 7. Схема взаимодействия Redux с Features

На рисунке 7 представлена модель взаимодействия Redux с компонентами, где хранится основная логика приложения – такие компоненты называются Features (основные компоненты, в которых сохраняется логика обработки запросов).

Заключение

Разработанная архитектура демонстрирует эффективное взаимодействие между компонентами системы умного дома. Использование современных технологий обеспечивает:

- Надежную защиту данных через HTTPS
 - Оперативную передачу информации посредством WebSocket
 - Гибкое хранение данных в MongoDB
 - Удобный интерфейс управления на базе React
- Система готова к масштабированию и может быть адаптирована под различные задачи автоматизации жилья.

Список литературы

1. Белов А. В. Практическая энциклопедия Arduino. М.: Наука и техника. ДМК Пресс, 2018. 272 с.
2. Блум Д. Изучаем Arduino Инструменты и методы технического волшебства: учебное пособие. М.: БХВ-Петербург, 2016. 336 с.
3. Геддес М. 25 крутых проектов с Arduino. М.: Эксмо, 2016.
4. Иго Т. Arduino, датчики и сети для связи устройств. М.: БХВ-Петербург, 2017. 544 с.
5. Володин В. Д., Шаронов А. А., Полевщиков И. С. Средства разработки и отладки программного обеспечения отечественных микропроцессорных устройств (часть 2) // Science Time. 2016. № 1(25). С. 91-94. EDN: VLIUFJ.

АНАЛИЗ СТАНДАРТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ И РАЗРАБОТКА ОПЕРАЦИОННОЙ МОДЕЛИ ДЛЯ ПРЕДПРИЯТИЙ КРИТИЧЕСКОЙ ИНФРАСТРУКТУРЫ

Мельников А. В., Мозговенко А. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@mozgovenko.ru

Задачи исследования:

1. Провести сравнительный анализ международных и национальных стандартов ИБ, применимых к КИ.

2. Выявить типовые угрозы и сценарии атак на объекты КИ (с учётом специфики энергетики, транспорта, ТЭК, связи и др.).

3. Определить ключевые требования регуляторов (ФСТЭК, ФСБ, Минцифры) к защите КИ.

Материалы и методы исследования

Современные исследования в области ИБ критической инфраструктуры фокусируются на следующих аспектах:

1. Регуляторная среда. Активно анализируются требования ФЗ-187, приказов ФСТЭК № 239 и № 31, а также международные стандарты (ISO/IEC 27001, NIST CSF, IEC 62443 для промышленных систем). Отмечается тенденция к гармонизации российских и зарубежных норм.

2. Киберугрозы для КИ. Публикации 2023–2025 гг. выделяют рост целевых атак (APT) на SCADA/ICS, использование вредоносного ПО типа TRITON/TRISIS, а также угрозы со стороны инсайдеров. Особое внимание уделяется уязвимостям устаревших промышленных протоколов (Modbus, DNP3).

3. Технологии защиты. Исследуются решения для:

- поведенческого анализа трафика (NTA);
- защиты конечных точек в промышленных сетях (EDR для ICS);
- автоматизации реагирования (SOAR);
- киберполигонов для тестирования устойчивости КИ.

4. Управление рисками. Развиваются методики количественной оценки рисков для КИ с учётом каскадных эффектов (например, отключение энергоснабжения ведёт к остановке транспорта).

Цель исследования – проанализировать операционную модель информационной безопасности для предприятий критической инфраструктуры, обеспечивающую соответствие регуляторным требованиям и устойчивую защиту от актуальных киберугроз.

Результаты исследования и их обсуждение

Для внедрения комплексной системы защиты информации и системы управления информационной безопасностью разработан ряд государственных и международных стандартов.

Обычно используют следующие стандарты для анализа, внедрения и постоянного мониторинга за СУИБ:

1. ГОСТ Р ИСО/МЭК 27001-2021 – российский аналог международного стандарта ISO/IEC 27001:2013. Устанавливает требования к созданию, внедрению, поддержанию и постоянному улучшению системы менеджмента информационной безопасности (СМИБ). Включает оценку и обработку рисков, выбор мер защиты, мониторинг эффективности и непрерывное совершенствование системы.

2. ГОСТ Р 50922-2006 – определяет основные термины и определения в области защиты информации. Важен для унификации понятий при разработке документации СУИБ.

3. ГОСТ Р ИСО/МЭК 15408 (три части: 1-2012, 2-2013, 3-2013) – «Общие критерии оценки безопасности информационных технологий». Определяет инструменты и методику оценки безопасности информационных продуктов и систем, включая критерии для сравнения результатов независимых оценок.

4. Стандарт Банка России СТО БР ИББС-1.0-2014 – применяется в банковской сфере. Регламентирует обеспечение информационной безопасности организаций банковской системы РФ, включая требования к защите данных и управлению рисками.

5. ГОСТ Р ИСО/МЭК 27013-2014 – руководство по совместному использованию стандартов ИСО/МЭК 27001 и ИСО/МЭК 20000-1 (менеджмент услуг). Помогает интегрировать требования к информационной безопасности и управлению услугами.

6. ГОСТ Р 51188-98 – регламентирует испытания программных средств на наличие компьютерных вирусов. Хотя стандарт устаревший, он может использоваться в части процедур тестирования ПО.

7. ГОСТ Р 51275-2006 – определяет факторы, воздействующие на информацию в объектах информатизации. Помогает в анализе угроз и уязвимостей.

В результате проведенного анализа государственных и международных стандартов по информационной безопасности, предлагаем использовать разработанную на основе вышеупомянутых стандартов операционную модель, которая состоит из ключевых аспектов, эффективная работа которых помогает обеспечить необходимый уровень ИБ в сочетании с бизнес-процессами.

Операционная модель включает следующие ключевые аспекты: политики, процессы ИБ, корпоративное управление, техническую архитектуру ИБ, организационную структуру, людей, ключевые показатели эффективности (КПЭ) и отчетность.

Процесс анализа операционной модели начинается с нормативной базы документов и процессов ИБ, а также оценки их уровня зрелости. Выполнение этого этапа необходимо для понимания отправной точки внедрения системы управления информационной безопасностью на предприятии.

Анализ существующей нормативной базы состоит из следующих шагов:

1. Собрать список документов, разработанных на предприятии и их номенклатуру

2. Проанализировать текущие политики и процедуры, связанные с информационной безопасностью

3. Проанализировать нормативные документы, основные нормы и правила соответствия требованиям регулятора и лучшим практикам (требования к предприятиям критической инфраструктуры, рекомендации ведущих практик ISO, NIST)

4. Определить недостающие документы и разработать перечень рекомендаций по дополнению существующей нормативной базы

Следующим этапом является оценка уровня зрелости установившихся процессов ИБ на предприятии. Для этого необходимо выполнить определенный перечень действий, а именно:

1. Проанализировать ключевые процессы ИБ

2. Оценить текущий уровень зрелости процессов ИБ и провести сравнительный анализ эффективности процессов с мировой и украинской практикой (бенчмаркинг)

3. Определить направления усовершенствования и развития процессов информационной безопасности



Модель процессов ИБ на предприятии

Оценка процессов ИБ в соответствии с их состоянием

0	Начальный: процесс отсутствует или его элементы выполняются частично
1	Выполняемый: основные элементы выполняются бессистемно. Процесс формально не задокументирован или документирован поверхностно, актуализация документации не выполняется. Эффективность функционирования зависит от индивидуальных знаний и усилий исполнителей
2	Установленный процесс: основные элементы процесса задокументированы и в основном выполняются одинаково для всей организации. Выполняется актуализация документации. Роли и обязанности определены и используются на практике. Процесс автоматизирован с помощью технических решений
3	Управляемый процесс: установленный процесс достигает поставленных результатов и генерирует показатели, на основе которых производится оценка его эффективности и усовершенствования
4	Оптимизированный процесс: высшее руководство принимает участие в анализе недостатков и повышении эффективности процесса. Совершенствование процесса и поддерживаемых технологий проводится регулярно и измеряется

Для оценки зрелости процессов ИБ по разным сферам предприятия, предлагаем использовать методологию для оценки зрелости процессов, разработанную на основе ведущих мировых практик и стандартов информационной безопасности. Для этого была создана референсная модель процессов ИБ по разным сферам предприятия (рисунок).

Были выделены следующие процессы:

- УД – Управление доступом к информационным ресурсам
- УИА – Управление информационными активами
- УПД – управление персональными данными
- УНБИС – Управление настройками безопасности ИС и оборудования
- УИНЦ- Управление инцидентами ИБ
- МБЗКТ – Сетевая безопасность и защита конечных точек
- МПИБ-мониторинг событий ИБ
- УВТС – Управление взаимодействием с третьими сторонами
- ЖБК – Обеспечение безопасности кода
- УВ – управление уязвимостями

Каждый процесс приведенной модели оценивается по пятибалльной шкале в соответствии с состоянием процесса на предприятии, толкование оценки процессов приведено в таблице.

Заключение

В этой статье рассмотрены государственные стандарты информационной безопасности, регламентирующие создание КСЗИ и международные стандарты информационной безопасности, что регламентируют построение СУИБ и определяют подход к построению системы управления кибербезопасностью. Были сравнительно государственные и международные стандарты на их соответствие современным тенденциям кибербезопасности.

Была разработана операционная модель, позволяющая систематизированно проанализировать текущее состояние организации в соответствии с требованиями вышеупомянутых

стандартов. Каждый раздел, входящий в операционную модель, был описан и приведен этапы проведения анализа для каждого из разделов. Представлен пример результатов анализа текущего состояния операционной модели предприятия, используя вышеупомянутые этапы и рекомендации.

Список литературы

1. Максимова Е. А. Аксиоматика инфраструктурного деструктивизма субъекта критической информационной инфраструктуры // Информатизация и связь. 2022. № 1. С. 68–74. DOI: 10.34219/2078-8320-2022-13-1-68-74. EDN: ZMOPQV.
2. Роберте Ф. С. Дискретные математические модели с приложениями к социальным, биологическим и экологическим задачам / пер. с англ. М.: Наука, 1986. 496 с.

КИБЕРБЕЗОПАСНОСТЬ. АНАЛИЗ ВИДОВ ЗАЩИТЫ В ИНФОРМАЦИОННОЙ СРЕДЕ

Пелипенко А. П.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: elena27712@mail.ru

Научный руководитель: Ступницкий В. С.

Введение

В нынешней цифровой среде обеспечение кибербезопасности приобретает первостепенное значение для устойчивого функционирования любой сферы жизни. Что же такое эта кибербезопасность? Это совокупность технических решений, направленных на защиту сетевых инфраструктур, хранимую информацию, устройств, обеспечение конфиденциальности и целостности. С каждым днём рост кибератак привлекает за собой новые виды угроз, а развитие систем защиты становится критически важным. Только в 2025 году было зарегистрировано более 36 фактов взлома с ущербом от 1 до 223 миллионов (USD). Цели кибератак могут быть разными: от кражи конфиденциальных данных, до нарушения работы систем.

Существует множество причин, по которым кибератаки стали такими частыми в наше время: быстрый рост технологий, развитие искусственного интеллекта, массовая цифровизация рабочих процессов, практически полный перенос хранения данных в виртуальные хранилища и облачные серверы. Эти факторы стимулируют злоумышленников разрабатывать новые и инновационные методы атак для достижения политических или коммерческих целей.

Цель исследования – на основе имеющихся данных сформировать целостную картину понимания о текущем состоянии угроз и найти методы противодействия кибератакам.

Материалы и методы исследования

Для достижения цели было проведено глубокое исследование с использованием отчетов ведущих международных агентств по кибербезопасности, статистические данные о кибератаках, актуальные исследования в области кибербезопасности. В ходе исследования были использованы: системный анализ, контент-анализ, сравнительный анализ.

Результаты исследования и их обсуждение

Количество кибератак стремительно усиливается, при этом большинство атак (70%) нацелены на нарушение работы или получение выкупа путём уничтожения данных. Значительная часть этих атак (44%) включает шифрование информации, 32% приводят к полному выводу из строя инфраструктуры. Злоумышленники, активно применяя ИИ для автоматизации, объединяются в группы и концентрируют свои усилия на промышленных, IT и финансовых организациях. Распространённые методы включают фишинг, социальную инженерию и использование уязвимостей веб-приложений. На рисунке 1 представлены актуальные киберугрозы за 2025 год.

Одним из самых распространённых угроз являются DDoS-атаки. В среднем ежедневно атакуется около 1 466 хостов, а в пиковые дни – до 8 532 хостов за сутки, общее число уникальных атакованных хостов за год превысило 483,9 тысячи, самая продолжительная атака длилась 10 дней, что указывает на растущую сложность и настойчивость злоумышленников.

Для защиты применяются комплексные решения (NGFW, EDR, XDR), SIEM-системы, шифрование, а также предиктивная аналитика и машинное обучение. Важнейшими мерами являются обучение сотрудников, внедрение принципа нулевого доверия, регулярное резервное копирование, постоянный мониторинг угроз и быстрое реагирование на инциденты, с учётом законодательных норм и стандартов кибербезопасности. На рисунке 2 представлены меры защиты от кибератак.

Разберём каждое решение и перспективы.

NGFW – является мощным средством защиты сетевой инфраструктуры, позволяющим выявлять многие современные угрозы. Одни из основных возможностей: контроль приложений, идентификация пользователей и устройств, предотвращение вторжений, песочница для файлов, обмен индикаторами, IPS. При грамотном внедрении NGFW, платформа повышает устойчивость и делает политику прозрачной.

EDR – это экспертная система для обнаружения и реагирования на современные угрозы на конечных устройствах (мобильные устройства, компьютер и т.д.). Включает защиту и сбор данных из любой конечной точки, анализ данных, уведомление о подозрительной активности, сохранение данных. Правильно настроенный EDR даёт возможность обнаруживать атаку на самых ранних этапах и отслеживать всю цепочку продвижения по инфраструктуре.

Следующим шагом в развитии технологий стало решение XDR. Это термин, который обозначает расширенное обнаружение и реагирование в сфере информационной безопасности. Его ценность проявляется в удобстве эксплуатации, повышенным уровнем автоматизации и эффективности поиска угроз и реагирования на них. Последовательно работает так: собирает и нормализует данные, анализирует и сопоставляет данные, содействует управлению инцидентами, помогает предотвратить будущие инциденты. XDR упрощает исследование и реагирование на операции безопасности за счёт объединения инструментов от нескольких поставщиков в единую экономичную платформу XDR. По мере роста внедрения, продолжается расширение возможностей данного решения. Одни из будущих трендов: объединение платформ, ИИ и автоматизация, аналитика поведения пользователей.

SIEM – система, обеспечивающая анализ в реальном времени событий безопасности. Основными задачами являются сбор, обработка и анализ событий безопасности, поступающих в систему из многочисленных источников. Немало важно обнаружение атак в режиме реального времени и выявление нарушений политик безопасности.

Шифрование – это способ преобразования данных, при котором они не смогут быть прочитаны кем-либо, кроме авторизованных сторон. Шифрование не обходится без криптографического ключа. Криптографический ключ – это уникальный набор символов, используемый для шифрования и дешифрования информации. Существует 2 вида: симметричный, при котором один и тот же ключ используется и для шифрования и для дешифрования. Асимметричный: используется пара разных ключей (открытый и закрытый).

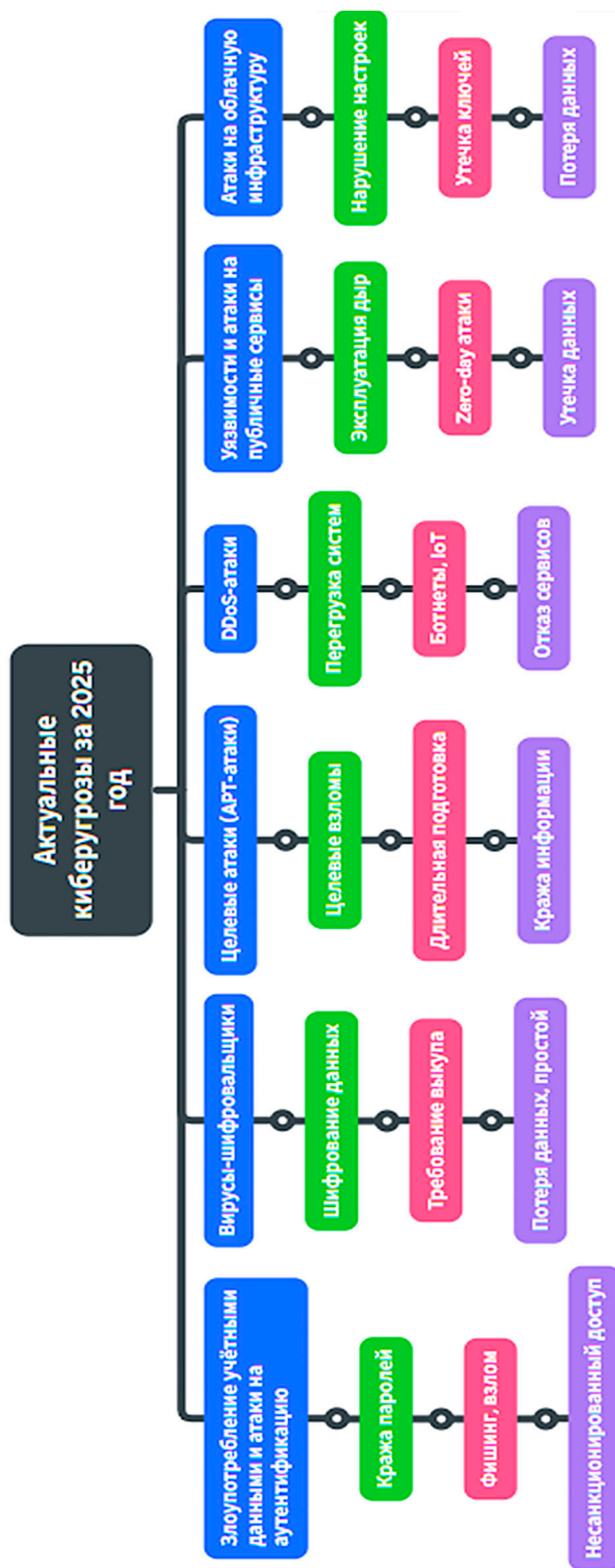


Рис. 1. Актуальные киберугрозы



Рис. 2. Меры защиты от кибератак

Предиктивная аналитика – это автоматизированный способ анализа данных для планирования и прогнозирования событий, который сопровождается алгоритмами машинного обучения и искусственного интеллекта. Используется в разнообразных областях бизнеса: здравоохранении, кредитовании, страховании и государственном секторе. Один из минусов: из-за существования многочисленных факторов, результаты предиктивной аналитики не всегда могут быть на 100 процентов достоверными.

Машинное обучение – отрасль искусственного интеллекта, позволяющая системам улучшать свою производительность без необходимости явного программирования. Оно включает в себя несколько этапов: сбор информации, подготовка данных, выбор алгоритма, обучение модели, оценка модели, доработка и внедрение. Машинное обучение открывает новые возможности для всех, кому нужно постоянно иметь дело с большими объёмами информации.

Заключение

Достижение 100 процентов безопасности на данный момент не представляется возможным, но объединив технологические средства, грамотную политику безопасности, осведомлённость пользователей, можно добиться эффективной защиты с области информационной безопасности.

Список источников.

1. Стратегический обзор киберугроз 2025 URL: <https://jetscirt.ru/analytics/kurs-na-antikhrupkost-strategicheskiiy-obzor-kiberugroz-2025/> (дата обращения: 15.12.2025).
2. Рик Ховард. Кибербезопасность главные принципы. 2024. URL: https://newsletter.radensa.ru/wp-content/uploads/2024/07/Кибербезопасность_главные_принципы_Рик_Ховард_2024.pdf (дата обращения: 15.12.2025).
3. Технология SIEM: полный обзор архитектуры, корреляции событий и интеграции с SOAR. URL: <https://serverflow.ru/blog/stati/tehnologiya-siem-polnyy-obzor-arkhitektury-korreljatsii-sobytyy-i-integratsii-s-soar/> (дата обращения: 15.12.2025).
4. Объединение силы и интеллекта: XDR как единое решение для надёжной киберзащиты 2023. URL: <https://www.securitylab.ru/analytics/544355.php> (дата обращения: 17.12.2025).

5. NGFW простыми словами. Как устроены современные межсетевые экраны и чем они отличаются от классических, 2024. URL: <https://www.securitylab.ru/analytics/563199.php> (дата обращения: 21.12.2025).

СТРАТЕГИЯ ПОВЫШЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В БАНКОВСКОМ ОТДЕЛЕНИИ

Поздняков И. В., Букреев Д. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru

Научный руководитель: Букреев Д.А.

Введение

Современная банковская деятельность в значительной степени опирается на использование информационных технологий, обеспечивающих обработку финансовых операций, хранение персональных данных клиентов и взаимодействие с государственными и межбанковскими информационными системами. В условиях цифровизации финансового сектора банковские отделения становятся не только точками обслуживания клиентов, но и важными элементами распределённой информационной инфраструктуры, от устойчивости и защищённости которых напрямую зависит надёжность функционирования кредитной организации в целом. Рост числа киберинцидентов в финансовой сфере, усложнение методов атак и повышение их целенаправленного характера актуализируют задачу перехода от фрагментарных мер защиты к стратегическому управлению информационной безопасностью. Банковские отделения, несмотря на ограниченные по сравнению с центральными офисами ресурсы, обрабатывают значительные объёмы конфиденциальной информации и нередко становятся уязвимым звеном в общей системе защиты. Это обусловлено сочетанием технических, организационных и человеческих факторов, включая использование автоматизи-

рованных рабочих мест, локальных сетей, периферийного оборудования и удалённых каналов доступа. Особенность обеспечения информационной безопасности в банковском отделении заключается в необходимости сочетать высокие требования к защите информации с непрерывностью бизнес-процессов и удобством обслуживания клиентов. Избыточно жёсткие меры безопасности способны снижать операционную эффективность, тогда как недостаточная защищённость создаёт предпосылки для утечек данных, финансовых потерь и репутационного ущерба. В этой связи особую значимость приобретает разработка стратегии информационной безопасности, ориентированной на системное управление рисками и адаптацию защитных мер к реальным условиям функционирования отделения.

Цель исследования – разработка и обоснование стратегии повышения уровня информационной безопасности в банковском отделении на основе анализа существующих угроз и уязвимостей.

Материал и методы исследования

В рамках исследования рассматривались автоматизированные рабочие места сотрудников, локальная вычислительная сеть отделения, используемое прикладное и системное программное обеспечение, а также регламенты обработки и защиты информации. Эмпирическую базу исследования составили нормативные и методические документы в области информационной безопасности банковской деятельности, требования регуляторов финансового сектора, а также открытые аналитические материалы, отражающие современные тенденции развития угроз и уязвимостей в кредитных организациях. Учитывались особенности функционирования банковского отделения как элемента распределённой инфраструктуры, взаимодействующего с центральными банковскими системами, платёжными шлюзами и внешними сервисами. Методологическая основа исследования базируется на сочетании аналитических и практико-ориентированных методов. В первую очередь применялся структурно-функциональный анализ, позволивший рассмотреть банковское отделение как совокупность взаимосвязанных подсистем – технической, программной, организационной и кадровой. Для оценки актуальных угроз и уязвимостей использовался риск-ориентированный метод, предполагающий идентификацию потенциальных источников угроз, анализ вероятности их реализации и оценку возможного ущерба. Это позволило перейти от формального соответствия требованиям безопасности к осмысленному управлению рисками, что является основой стратегического подхода к защите информации.

Результаты исследования и их обсуждение

Текущее состояние информационной безопасности банковского отделения формируется под воздействием совокупности технических, организационных и человеческих факторов, каждый из которых в отдельности может не представлять критической угрозы, однако в совокупности способен существенно снизить общий уровень защищённости. В условиях распределённой банковской инфраструктуры отделение выступает как автономный, но тесно связанный с центральными системами элемент, что делает его одновременно уязвимым и значимым с точки зрения обеспечения устойчивости всей информационной системы банка.

Особенность анализа информационной безопасности на уровне отделения заключается в необходимости учитывать реальные условия эксплуатации: ограниченные вычислительные ресурсы, высокую нагрузку на персонал, интенсивный поток клиентов и жёсткие требования к непрерывности обслуживания [1]. В таких условиях меры защиты зачастую реализуются фрагментарно и ориентированы преимущественно на выполнение формальных требований, что снижает их эффективность при возникновении нестандартных ситуаций и целевых атак.

1. Организационная составляющая системы информационной безопасности банковского отделения играет ключевую роль, поскольку именно она определяет порядок применения технических средств защиты и поведение персонала в различных ситуациях. Анализ показывает, что в ряде случаев политика информационной безопасности носит декларативный характер и недостаточно адаптирована к условиям конкретного отделения. Регламенты и инструкции по защите информации часто разрабатываются на уровне головного офиса и не учитывают специфику локальных бизнес-процессов. В результате сотрудники воспринимают требования безопасности как формальные ограничения, что приводит к их частичному игнорированию или формальному выполнению. Недостаточная регламентация ответственности и отсутствие чётких процедур реагирования на инциденты дополнительно усиливают риски, связанные с человеческим фактором.

2. Программно-аппаратная инфраструктура банковского отделения, как правило, включает автоматизированные рабочие места сотрудников, локальную вычислительную сеть, серверные компоненты и периферийные устройства. Анализ состояния этих элементов показывает, что основными источниками уязвимостей являются неоднородность используемого оборудования и программного обеспечения, а также несвоевременное обновление средств защиты. В условиях ограниченных ресурсов часть рабочих станций может эксплуатироваться дли-

тельное время без модернизации, что повышает вероятность использования уязвимостей операционных систем и прикладных программ. Средства антивирусной защиты и межсетевые экраны нередко функционируют в стандартных конфигурациях, не адаптированных под конкретные угрозы, характерные для банковского сектора. Это снижает их способность эффективно противодействовать целенаправленным атакам и современным вредоносным программам.

3. Человеческий фактор остаётся одним из наиболее значимых источников рисков информационной безопасности в банковском отделении. Сотрудники ежедневно работают с конфиденциальной информацией и взаимодействуют с клиентами, что создаёт благоприятные условия для реализации методов социальной инженерии. При этом уровень осведомлённости персонала в вопросах информационной безопасности зачастую оказывается недостаточным для своевременного распознавания угроз. Отсутствие регулярного обучения и практических тренингов приводит к тому, что сотрудники не всегда осознают последствия своих действий, связанных с обработкой информации и использованием информационных систем. Использование простых паролей, передача учётных данных, работа с подозрительными электронными сообщениями – все эти факторы повышают вероятность компрометации данных и нарушений безопасности.

В целом анализ текущего состояния информационной безопасности банковского отделения показывает, что существующие меры защиты не всегда образуют единую и согласованную систему. Фрагментарность организационных решений, технические ограничения и влияние человеческого фактора создают предпосылки для возникновения инцидентов безопасности. Это обуславливает необходимость перехода от локальных корректирующих мер к разработке и реализации целостной стратегии повышения информационной безопасности.

Разработка стратегии повышения информационной безопасности банковского отделения должна опираться на результаты комплексного анализа текущего состояния защищённости и учитывать, как внутренние особенности функционирования отделения, так и внешние регуляторные и технологические требования. В отличие от набора разрозненных защитных мер, стратегия предполагает системный и долгосрочный подход, ориентированный на управление рисками, адаптацию к изменяющейся угрозой обстановке и устойчивое развитие системы безопасности.

Стратегический подход позволяет перейти от реагирования на отдельные инциденты к проактивному управлению информационной безопасностью. В условиях банковской деятельности это особенно важно, поскольку последствия нарушений безопасности выходят за рамки ло-

кальных потерь и могут затрагивать финансовую устойчивость, репутацию и доверие клиентов к кредитной организации.

В основе стратегии повышения информационной безопасности банковского отделения лежит ряд базовых принципов, определяющих логику принятия решений и выбор защитных мер:

1. К числу ключевых принципов относится приоритет риск-ориентированного подхода, при котором ресурсы направляются на защиту наиболее критичных активов и процессов. Такой подход позволяет обеспечить рациональное использование средств и избежать избыточных мер, не приносящих существенного повышения уровня защищённости.

2. Не менее важным является принцип согласованности стратегии информационной безопасности с бизнес-целями банковского отделения. Защитные меры не должны препятствовать выполнению основных функций обслуживания клиентов и обработки финансовых операций. Напротив, стратегия должна способствовать повышению устойчивости бизнес-процессов и снижению вероятности их нарушения вследствие инцидентов безопасности [2].

3. Дополнительным принципом выступает адаптивность стратегии, предполагающая её регулярный пересмотр с учётом изменения угроз, технологий и нормативных требований. В условиях быстрого развития цифровых финансовых сервисов статическая модель защиты оказывается неэффективной, что требует постоянного мониторинга и корректировки принятых решений.

Реализация стратегии повышения информационной безопасности в банковском отделении предполагает выделение нескольких ключевых направлений, каждое из которых ориентировано на снижение определённой группы рисков. Одним из таких направлений является совершенствование организационно-управленческой структуры информационной безопасности. Чёткое распределение ролей и ответственности, актуализация внутренних регламентов и формирование процедур реагирования на инциденты создают основу для эффективного применения технических средств защиты.

Важным стратегическим направлением является модернизация программно-аппаратной инфраструктуры с учётом актуальных угроз. Это включает унификацию используемых средств защиты, централизованное управление обновлениями и настройками, а также внедрение механизмов контроля целостности и мониторинга состояния информационных систем. Такой подход позволяет снизить вероятность эксплуатации уязвимостей, связанных с устаревшими или некорректно настроенными компонентами.

Отдельного внимания заслуживает развитие системы управления доступом и аутентификации. В условиях банковского отделения, где сотрудники имеют доступ к различным уровням информа-

ции, стратегически важно обеспечить принцип минимально необходимых привилегий и предотвратить несанкционированное расширение прав доступа [2]. Это снижает риски как внешних атак, так и внутренних нарушений безопасности.

Эффективность стратегии информационной безопасности во многом определяется степенью её интеграции в общую систему управления банковским отделением. Стратегия не должна существовать изолированно в виде формального документа [5], а должна быть встроена в процессы планирования, контроля и оценки деятельности отделения.

Интеграция стратегии предполагает регулярную оценку уровня информационных рисков, использование показателей эффективности защитных мер и проведение внутренних аудитов безопасности. Это позволяет не только выявлять слабые места, но и оценивать результативность реализуемых мероприятий. Важную роль играет также обратная связь от персонала, позволяющая учитывать практические аспекты применения защитных мер и корректировать стратегию с учётом реальных условий эксплуатации.

Таким образом, формирование стратегии повышения информационной безопасности банковского отделения представляет собой многоэтапный процесс, ориентированный на системное управление рисками и устойчивость информационной инфраструктуры [3]. Заложенные в стратегии принципы и направления создают основу для практической реализации конкретных мероприятий, которые рассмотрим более конкретно. В отличие от разрозненных технических решений, практическая реализация стратегии ориентирована на согласованное развитие организационных, технических и кадровых компонентов системы безопасности, что позволяет обеспечить их взаимное усиление и устойчивость к изменяющимся условиям. Практическая ценность стратегии проявляется именно на уровне повседневной деятельности отделения, где защитные меры должны быть встроены в рабочие процессы и восприниматься персоналом как неотъемлемая часть профессиональной деятельности, а не как внешнее ограничение.

Одним из ключевых направлений практической реализации стратегии является модернизация технических и программных средств защиты информации. В условиях банковского отделения приоритет отводится унификации и централизации средств защиты, что позволяет снизить вероятность ошибок конфигурации и упростить администрирование.

1. Практическая реализация данного направления включает внедрение централизованного управления антивирусной защитой и средствами контроля целостности, а также регулярное обновление операционных систем и прикладного программного обеспечения [4]. Особое внимание уделяется защите автоматизированных рабочих

мест сотрудников, которые являются основной точкой взаимодействия с банковскими информационными системами и, одновременно, одним из наиболее уязвимых элементов инфраструктуры.

2. Дополнительно в рамках стратегии предусматривается усиление сетевой защиты, включая актуализацию правил межсетевого экранирования, контроль удалённого доступа и мониторинг сетевого трафика. Такие меры позволяют своевременно выявлять аномальную активность и предотвращать развитие инцидентов безопасности на ранней стадии.

3. Организационные меры играют системообразующую роль в реализации стратегии информационной безопасности. Даже наиболее совершенные технические решения теряют эффективность при отсутствии чётко выстроенных регламентов и осознанного участия персонала. В этой связи стратегия предусматривает актуализацию внутренних нормативных документов, регламентирующих порядок работы с информацией, использование средств защиты и реагирование на инциденты.

Практическая реализация организационных мер включает проведение регулярных инструктажей и обучающих мероприятий для сотрудников банковского отделения. Обучение ориентировано не только на формальное ознакомление с требованиями безопасности, но и на формирование понимания возможных последствий нарушений и роли каждого сотрудника в обеспечении защиты информации. Особое внимание уделяется управлению доступом и ответственности персонала. Чёткое разграничение прав, фиксация действий пользователей и контроль соблюдения регламентов позволяют снизить риски, связанные с внутренними нарушениями и человеческим фактором. В рамках стратегии также предусматривается формирование культуры информационной безопасности, при которой соблюдение требований защиты воспринимается как профессиональный стандарт.

Оценка эффективности реализуемых мер осуществляется на основе анализа инцидентов безопасности, результатов внутренних проверок и показателей информационных рисков. Такой подход позволяет выявлять слабые места и корректировать стратегию без ожидания серьёзных нарушений. В условиях банковского отделения особую значимость приобретает оперативность такой оценки, поскольку задержки в принятии решений могут привести к существенным финансовым и репутационным потерям.

Адаптация стратегии предполагает не только обновление технических средств защиты, но и пересмотр организационных подходов, программ обучения персонала и механизмов контроля. Это обеспечивает гибкость системы информационной безопасности и её соответствие актуальным условиям функционирования банковского отделения.

Заключение

Анализ текущего состояния информационной безопасности выявил, что основными источниками рисков на уровне банковского отделения являются фрагментарность организационных решений, неоднородность программно-аппаратной инфраструктуры и влияние человеческого фактора. Формальное выполнение требований безопасности без их адаптации к реальным условиям функционирования отделения снижает эффективность защитных мер и не позволяет своевременно реагировать на изменение угрозой обстановки. Сформированная стратегия повышения уровня защищённости ориентирована на согласование мер безопасности с бизнес-процессами банковского отделения, обеспечение адаптивности системы защиты и её интеграцию в процессы управления и контроля. Такой подход позволяет не только снизить вероятность реализации актуальных угроз, но и повысить устойчивость информационной инфраструктуры в долгосрочной перспективе. Модернизация средств защиты, развитие системы управления доступом, повышение уровня осведомлённости персонала и регулярная оценка эффективности принятых решений позволяют сформировать целостную и управляемую систему информационной безопасности на уровне банковского отделения. Таким образом, стратегия повышения информационной безопасности банковского отделения должна рассматриваться как непрерывный процесс, направленный на управление рисками, адаптацию к изменяющимся условиям и поддержание доверия клиентов.

Список литературы

1. Столлинге У. Криптография и защита сетей: принципы и практика. М.: Вильямс, 2020.
2. Шнайер Б. Прикладная криптография. М.: Диалектика, 2019.
3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2021.
4. Pfleeger C., Pfleeger S., Margulies J. Security in Computing. Pearson, 2020.
5. ISO/IEC 27001:2022. Information Security Management Systems – Requirements.

МОДЕЛЬ ЯСТРЕБ-ГОЛУБЬ ДЛЯ АНАЛИЗА АГРЕССИВНОГО И ДРУЖЕЛЮБНОГО ВЕБ-ИНТЕРФЕЙСА

Попов Т. И., Покуса Т. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: tamilapokusa@mail.ru*

Научный руководитель: Покуса Т. В.

Введение

Современный интернет имеет крайне широкий спектр выбора для пользователей среди сайтов предоставляющие услуги. Регулярно случа-

ются инциденты, когда интернет-пользователи могут столкнуться с мошенническим сайтом, или как минимум – с сайтом использующие человеческие качества пользователя для того, чтобы повысить конверсию своей площадки. Такой подход зачастую агрессивен с точки зрения дизайна и маркетинга, с другой стороны существуют сайты с повышенной прозрачностью и вызовом доверия у пользователей, но они парадоксально имеют куда меньший охват пользователей, чем сайты с агрессивным подходом. С точки зрения теории игр, эти две категории сайтов можно поделить на две группы: «Ястребы» и «Голуби». Согласно математической модели «Ястребы и голуби», её смысл состоит в описывании конкурентных отношений в некоторой популяции животных и выработки оптимальной эволюционной стратегии. Данная модель будет адаптирована для экосистем рынка веб-сайтов: «Ястреб» – сайты использующие методы скрытого воздействия в дизайне и громкую рекламу, «Голуби» – сайты менее заметные, чем «ястребы» из-за меньшей рекламной политики и имеющие большую степень честности по отношению к покупателям. Используя вышеуказанную модель из теории игр, в данной научной работе будет исследоваться потенциально наиболее выгодная модель для рынка и при каких условиях рынок может достичь равновесия в конкуренции «голубей» и «ястребов». Практическая значимость состоит в том, что формализация конфликта даёт чёткие критерии, по которым пользователи распознают манипулятивные сайты. Модель позволяет владельцам площадок и аналитикам проверить, насколько выгодна та или иная стратегия при сложившихся условиях рынка, а также определить долю «ястребов» и «голубей», при которой рынок переходит в устойчивое состояние.

Цель исследования – нахождение точки равновесия в интернет-рынке для сайтов из групп «ястребов» и «голубей». Вычисление наиболее оптимальной стратегии поведения для веб-сайтов. Найти способ распознавать сайты использующие манипуляции.

Материалы и методы исследования

В работе используется метод из теории игр «Ястреб и голубь». Джон Мейнард Смит описал ситуацию, в которой животные выбирают: уступить или драться. Это вариант игры «Струсил – проиграл». В споре за добычу оба соперника обычно проявляют агрессию и стремятся ранить друг друга. Непосредственно перед схваткой каждый решает: отступить, оставив добычу, но оставившись живым (голубь), или продолжать бой до конца, рискуя погибнуть (ястреб). Предположим, в популяции, где все ведут себя как голуби, появляется небольшая партия ястребов. Сначала доля ястребов увеличивает-

ся, потому что их поведение приносит выгоду: при встрече голубя и ястреба побеждает последний. По мере того как ястребов становится больше, сталкиваются они чаще, а потому травмы между ними учащаются. В конце концов устанавливается устойчивое соотношение между ястребами и голубями. В природе наблюдается именно такой баланс. С учётом сказанного, была создана платёжная матрица Смитом. Платежи были бы определены следующим принципом: достижение цели (V) оценивается в 10 очков, увечья (W) приносят – 20 очков. В схватке ястребов между собой, победа и поражение чередуются и каждый получает – 5 очков, в схватке ястреба и голубя, ястреб всегда побеждает и получает 10 очков, а голубь отступает (L) и получает 0 очков. В схватке голубей пострадавших нет, но голуби тратят время (E), оценка ситуации в – 3 очка. В голубиной схватке победитель получает $10 - 3 = 7$ очков, проигравший – 3 очка, в среднем каждый получает по 2 очка [1].

	Ястребы	Голуби
Ястребы	(-5; -5)	(10; 0)
Голуби	(0; 10)	(2; 2)

Существуют формулы для вычисления платёжной матрицы:

$$S(\Gamma, \Gamma) = ((V - L) / 2) - E;$$

$$S(\mathcal{A}, \mathcal{A}) = (V - W) / 2;$$

$$S(\Gamma, \mathcal{A}) = -L;$$

$$S(\mathcal{A}, \Gamma) = V.$$

Также будут в научной работе будут использоваться формулы для вычисления:

среднего выигрыша ястребов

$$S_H(z) = z \times S(\mathcal{A}, \mathcal{A}) + (1 - z) \times S(\mathcal{A}, \Gamma),$$

среднего выигрыша голубей

$$S_D(z) = z \times S(\Gamma, \mathcal{A}) + (1 - z) \times S(\Gamma, \Gamma),$$

нахождения доли ястребов

$$z = \frac{V + L + 2E}{W - L + 2E}.$$

Метод «Ястреб и голубь» требует адаптации под цель научной работы – анализ веб-сайтов, поэтому для поиска переменных нельзя в точности соблюдать классическому методу. 1) Переменная V – конверсия сайта, находится путём подсчёта решений через таблицу в дизайне нацеленных на повышение конверсии сайта, каждая реализация добавляет сайту 1 балл, отсутствие решения даёт 0 баллов. 2) Переменная W – методы скрытого воздействия, находится аналогичным образом, подсчитываются элементы интерфейса, что используют манипулятивные техники над пользователями для дости-

жения своих целей. Дополнительно, в таблице чёрных паттернов объясняется опасность того или иного решения с ссылкой на книгу о манипулятивных приёмах в дизайне – «Evil by design» за авторством Криса Ноддера [2]. 3) Переменная E – потраченное время пользователя для принятия решения, является когнитивной нагрузкой на покупателя, чем больше времени уходит у человека, чтобы считать элементы сайта, тем выше вероятность, что он его покинет. Когнитивная нагрузка высчитывается по закону Хика. Закон Хика, также называемый законом Хика–Хаймана, назван в честь британского психолога Уильяма Эдмунда Хика и американского психолога Рэя Хаймана. Он описывает, как растёт время, требующееся человеку для выбора, при добавлении новых альтернатив: чем больше появляется возможных вариантов, тем дольше приходится ждать, пока человек примет решение, причём это увеличение следует логарифмическому закону. Закон Хика имеет следующую формулу

$$L = b \times \log_2(n + 1),$$

где b – константа, которая может быть определена эмпирически путем подгонки линии к измеренным данным. Добавление 1 к n учитывает «неопределенность относительно того, отвечать или нет, а также относительно того, какой ответ дать» [3]. На место константы b , будет ставиться конкретное значение из исследования «Making Up Your Mind After a 100-Ms Exposure to a Face» за авторством Дженина Уиллиса и Александра Тодорова [4]. В исследовании доказывалось научным методом, что людям хватает 100 миллисекунд, чтобы сформировать мнение о человеке, а увеличение время просмотра на лицо, в большей степени, лишь закрепляет мнение человека [4]. Таким образом, 100 миллисекунд равняются 0.1 секунде и данное значение будет использоваться как значения b в формуле. Вместо значения n подставляется общая сумма интерактивных элементов на сайте, что подсчитываются визуально. Стоит учесть, что исследования Хика не является полностью объективным определением влияния сложности восприятия объекта на реакцию человека. Закон Хика, трактуют время реакции (Reaction Time – RT) как прямое отражение длительности когнитивных процессов, современные исследования ставят эту трактовку под сомнение. Работа Wong A. L. (2017) показывает, что RT может быть в значительной степени подвержен опытозависимому смещению, функционируя как независимый параметр движения, устанавливаемый по привычке [5]. Это означает, что увеличение времени реакции в условиях множественного выбора может отражать не только рост вычислительной сложности, но и формирование адаптивной двигательной стратегии в ответ на неопреде-

ленность. Таким образом, интерпретация данных RT, в том числе в контексте закона Хика, требует учёта истории двигательного опыта испытуемого и допущения, что время реакции несёт в себе как вычислительную, так и привычную компоненту [5]. Тем не менее, закон Хика всюду используется современными дизайнерами и маркетологами для оценки сложности восприятия сайтов, по данной причине, нахождение переменной E будет проводиться данным путём. 4) Значение L из метода «ястреба и голубя» не находится, и всегда равняется 0. Само исследование происходит следующим образом.

Для реализации методологии анализа дизайн-стратегий формируются две группы сайтов: группа «голубь» и группа «ястреб», по два сайта в каждой. Для повышения репрезентативности сравнения каждый исследуемый сайт будет парно сопоставляться с прямым или косвенным конкурентом из противоположной группы, действующим на одном рыночном сегменте. Это позволит минимизировать влияние отраслевой специфики на результаты оценки переменных V, W и E. Экспертная оценка каждого сайта по всем критериям будет проводиться независимо, после чего полученные баллы усредняются внутри каждой группы. Это даёт усреднённые групповые значения $V_{\text{группа}}, W_{\text{группа}}, E_{\text{группа}}$, отражающие типичные характеристики стратегий «голубь» и «ястреб». Для построения единой платёжной матрицы, описывающей взаимодействие в общей популяции сайтов, необходимо рассчитать общие средние значения переменных V, W и E по всей выборке. Эти финальные значения $V_{\text{общее}}, W_{\text{общее}}, E_{\text{общее}}$, наряду с константой $L = 0$, и подставляются в формулы Смита для расчёта выигрышей в клетках матрицы:

$$S(\Gamma, \Gamma) = ((V_{\text{общ}} - L) / 2) - E_{\text{общ}};$$

$$S(\mathcal{A}, \mathcal{A}) = (V_{\text{общ}} - W_{\text{общ}}) / 2;$$

$$S(\Gamma, \mathcal{A}) = -L;$$

$$S(\mathcal{A}, \Gamma) = V_{\text{общ}};$$

Именно на основе этой общерыночной матрицы затем вычисляется средний выигрыш ястребов $S_H(z)$ и голубей $S_D(z)$ в зависимости от доли ястребов z, а также находится точка равновесия z^* , при которой выгоды от обеих стратегий сравниваются и рынок достигает устойчивого состояния.

Результаты исследования и их обсуждение

Основа исследования собрана из четырёх веб-сайтов. Два из них – онлайн-кинотеатры, два других – сервисы пополнения счёта в игровой платформе Steam. В каждой паре выбраны конкурирующие ресурсы с противоположными подходами к пользователю. Среди кинотеатров сайт Ivi показывает агрессивную стратегию «Ястреба», а «Кинопоиск» – честную стратегию «Голубя». Среди платёжных сервисов поведение «Ястреба» иллюстрирует Kupikod, а «Голубя» – PayBerry. Первый качественный этап оценил переменные модели. Для пары Ivi и «Кинопоиск» в таблице 1 собраны скрытые манипулятивные приёмы, распределённые по типологии Криса Ноддера в книге «Evil by Design» [2]. Эти данные дали значение переменной W – потенциал негатива. В таблице 2 для тех же сайтов перечислены дизайн-решения, повышающие конверсию; по ним вычислена переменная V – ценность ресурса. Аналогичный разбор проведён для пары Kupikod и PayBerry. Заключительный этап формализует собранные данные в модели «Ястреб и Голубь»: вычисляются средние значения, строится платёжная матрица и находится точка равновесия. «Ястреба и голубя».

Таблица 1

Анализ веб-сайтов «Ivi» и «Кинопоиск» в области онлайн-кинотеатров

№	Скрытая манипуляция	Кинопоиск	Ivi
1	Принудительная непрерывность и сложная отмена	1	1
2	Скрытая информация	0	1
3	Принудительный конструктор	1	1
4	Имитация срочности	0	1
5	Имитация редкости	0	1
6	Скрытое подтверждение	0.	1
7	Социальное доказательство	0	1
8	Привязка к аккаунту	0	1
9	Навязчивые уведомления	1	1
10	Плата за удобство	1	1

Кинопоиск: $W = 4$;

Ivi: $W = 10$.

№	Дизайн-решение, повышающее конверсию	Кинопоиск	Ivi
1	Бесплатный просмотр первой серии	1	1
2	Автовоспроизведение следующей серии	1	1
3	Массовая реклама	1	1
4	Эмоциональные постеры	1	1
5	Чёткое разделение бесплатного и платного	1	0
6	Пакетные предложения (бандлы)	1	0
7	Агрессивные уведомления о премьерях	0	1
8	Многоуровневая система подписок	0	1
9	Персонализированные рекомендации	1	1
10	Единый аккаунт с экосистемой	1	0

Кинопоиск: $V = 8$;

Ivi: $V = 7$.

Кинопоиск: $E = 0.1 \times \log_2(8+1) = 0.316$ секунды.

Ivi: $E = 0.1 \times \log_2(11+1) = 0.358$ секунды.

Таблица 2

Анализ сайтов «Kupikod» и «PayBerry» в области пополнения аккаунта для игрового магазина «Steam»

№	Скрытая манипуляция	Kupikod	PayBerry
1	Скрытые дополнительные платежи	1	0
2	Искусственное создание ажиотажа	1	0
3	Затруднённая процедура отказа от услуг	1	0
4	Автоматическое подключение платных опций	1	0
5	Использование внутренней виртуальной валюты	1	0
6	Установка завышенного ценового ориентира	1	0
7	Привлечение авторитетных мнений	1	0
8	Система поощрений за совершение действий	1	1
9	Иллюзия выгоды от увеличения суммы	1	0
10	Неисполнение обязательств после оплаты	1	0

Kupikod: $W = 10$;

PayBerry: $W = 1$.

№	Дизайн-решения повышающие конверсию	Kupikod	Payberry
1	Активное саморекламирование	1	0
2	Возможность совершить транзакцию с главной страницы	1	0
3	Реклама самых выгодных условий	1	0
4	Подключение к подписке после первой покупки	1	0
5	Визуализация этапов заполнения данных необходимых для транзакции	1	0
6	Подробные инструкции как совершить транзакцию	1	1
7	Снижении комиссии при повышении суммы совершаемой транзакции	1	0
8	Чистый и понятный дизайн	0	1
9	Отображение снимаемой суммы с учётом комиссии	1	1
10	Получение чека	0	1

Kupikod: $V = 8$;

PayBerry: $V = 4$.

Kupikod: $E = 0.1 \times \log_2(6+1) = 0.281$ секунды.

PayBerry: $E = 0.1 \times \log_2(20+1) = 0.439$ секунды.

3. Вычисление результатов по методу «Ястреба и голубя» используя полученные данные.

Группа «голубей»:

Кинопоиск ($V = 8; W = 4; E = 0.316$),

PayBerry ($V = 4; W = 1; E = 0.2807$).

Группа «ястребов»:

Ivi ($V = 7; W = 10; E = 0.358$),

Kurikod ($V = 8; W = 10; E = 0.439$).

Средние по группе «Голубь»:

$$V = \frac{(8+4)}{2} = 6.0$$

$$W = \frac{(4+1)}{2} = 2.5$$

$$E = \frac{(0.281+0.316)}{2} = 0.2985$$

Средние по группе «Ястреб»:

$$V = \frac{(7+8)}{2} = 7.5$$

$$W = \frac{(10+10)}{2} = 10.0$$

$$E = \frac{(0.439+0.358)}{2} = 0.3985$$

Общерыночные средние (ключевые для матрицы):

$$V = \frac{(6.0+7.5)}{2} = 6.75$$

$$W = \frac{(2.5+10.0)}{2} = 6.25$$

$$E = \frac{(0.2985+0.3985)}{2} = 0.3485$$

Перевод общих средних значений в баллы модели:

$$V = \frac{6.75}{10} \times 10 = 6.75;$$

$$W = \left| -\frac{6.25}{10} \right| \times 20 = 12.5;$$

$$E = -\left(\frac{0.3485}{0.439} \right) \times 3 = -2.38;$$

$$L = 0;$$

Построение платёжной матрицы:

$$S(G, G) = \frac{6.75}{2} - (-2.38) = 5.755;$$

$$S(Y, Y) = \frac{6.75 - 12.5}{2} = -2.875;$$

$$S(G, Y) = 0;$$

$$S(Y, G) = 6.75;$$

	Ястребы	Голуби
Ястребы	(-2.875; -2.875)	(6.75; 0)
Голуби	(0; 6.75)	(5.755; 5.755)

Нахождение равновесия доли ястребов (Z):

$$z = \frac{6.75 + 0 + 2 * (-2.38)}{12.5 - 0 + 2 * (-2.38)} = 0.257$$

$$S_Y(z) = 0.257 * (-2.875) + (1 - 0.257) * 6.75 = 4.276$$

$$S_G(z) = 0.257 * 0 + (1 - 0.257) * 5.755 = 4.275 .$$

Заключение

Исследование адаптировало стандартную игровую модель «ястреб и голубь» для анализа конкуренции между веб-сайтами. В работе выделили четыре переменные – V, W, E, L – и показали, как оценить каждую: дизайн-эксперт задаёт значения, а когнитивную нагрузку вычисляются по закону Хикса. Методику проверили на четырёх сайтах, сгруппированных в две пары-конкурента: одна пара из онлайн-кинотеатров, вторая из сервисов цифровых платежей. Эксперты зафиксировали чёткое разделение стратегий: «ястребы» скрыто манипулируют пользователем через тёмные паттерны, а «голуби» делают интерфейс прозрачным и убирают лишние шаги. Анализ получившихся среднерыночных параметров ($V = 6.75; W = 12.5; E = -2.38, L = 0$) и построенная на их основе платёжная матрица показали эволюционно-стабильное состояние рынка. Модель выдала, что равновесие наступает тогда, когда доля площадок, выбравших агрессивную стратегию «ястреб», равняется $z^* \approx 25.7\%$, а оставшиеся 74.3% придерживаются честной стратегии «голубь». В этой точке средний выигрыш каждой стратегии одинаков и равен примерно 4.28 условных единиц. Результат важен как для теории, так и для практики. В теории он показывает: даже при заметном конверсионном преимуществе агрессивных сайтов ($V_{\text{ястреб}}$ выше $V_{\text{голубь}}$), их рост тормозится большими издержками взаимной конкуренции. Когда встречаются два «ястреба», выигрыш становится отрицательным (-2.875), поэтому у «шолубей» формируется устойчивая экологическая ниша. В практическом плане рассчитанная доля агрессоров (25.7%) меньше той, что зафиксирована в выборке (50%), значит при сохранении текущих условий рынок может сместиться к более честным практикам. Модель формализует конфликт интересов и даёт численную базу для прогноза дизайн-стратегий: в долгосрочной перспективе стратегия

«голубь» не только этически предпочтительна, но и эволюционно устойчива, занимая большую часть равновесия.

Список литературы

1. Деулофеев Х. Теория игр. Дилемма заключённого и доминантные стратегии. Москва: Ленанд, 2014. 272 с.
2. Nodder C. *Evil by Design: Interaction Design to Lead Us into Temptation*. Indianapolis: John Wiley & Sons, 2013. 258 p.
3. Hick W. E. On the rate of gain of information // *Quarterly Journal of Experimental Psychology*. 1952. Vol. 4. No. 1. P. 11–26.
4. Willis J., Todorov A. First Impressions: Making Up Your Mind After a 100-ms Exposure to a Face // *Psychological Science*. 2006. Vol. 17. No. 7. P. 592–598.
5. Wong A. L., Goldsmith J., Forrence A. D., Haith A. M., Krakauer J. W. Reaction times can reflect habits rather than computations // *eLife*. 2017. Vol. 6. Article e28075. P. 1–18. DOI: 10.7554/eLife.28075.

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ JAVA И C++ ПРИ СОРТИРОВКЕ ЦЕЛЫХ ЧИСЕЛ И ЗАПИСИ/ЧТЕНИИ ФАЙЛОВ

Романенко И. В.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ryukjwfr@gmail.com

Научный руководитель: Ступницкий В. С.

Введение

Выбор языка программирования оказывает существенное влияние на производительность, надёжность и сопровождаемость программного обеспечения. Особенно важен этот выбор при разработке приложений, требующих эффективной обработки данных и выполнении базовых операций, таких как сортировка и работа с файлами. Среди множества языков особое внимание заслуживают C++ и Java, два объектно-ориентированных языка, широко применяемых в промышленной разработке, но реализованных на разных уровнях системной абстракции. C++ компилируемый язык, близкий к аппаратному уровню, предоставляющий разработчику полный контроль над ресурсами системы. Однако эта гибкость сопряжена с повышенной сложностью: ошибки в управлении памятью (например, утечки или двойное освобождение) могут привести к нестабильной работе. Этот недостаток можно смягчить за счёт строгого соблюдения правил RAII (Resource Acquisition Is Initialization) и использования современных средств языка, таких как автоматические локальные переменные и контейнеры стандартной библиотеки (например, `std::vector`), которые управляют памятью самостоятельно. Java, напротив, выполняется в среде виртуальной машины и обеспечивает автоматическое управление памятью с помощью сборщика мусора. Это повышает надёжность и упрощает разработку, но вносит накладные расходы, что может сни-

зить производительность в вычислительных интенсивных задачах. Этот недостаток так же можно частично компенсировать настройкой параметров JVM и применением пулов памяти или off-heap решений. Эти особенности делают сравнение языков актуальным: несмотря на схожесть в синтаксисе и парадигмах, они по-разному влияют на эффективность выполнения одних и тех же операций. Ранние исследования (например, Prechelt, 1999; Nikishkov et al., 2003) отмечали значительное отставание Java по скорости и потреблению памяти. Однако современные версии компиляторов и виртуальных машин существенно улучшили ситуацию, что требует актуального повторного сравнения в контексте типовых задач.

Цель исследования – провести сравнительный анализ производительности языков программирования C++ и Java при выполнении базовых операций, таких как сортировка целых чисел, чтение и запись файлов, с целью выявления их относительных преимуществ и недостатков в типичных сценариях использования.

Материалы и методы исследования

Для сравнения производительности языков C++ и Java были созданы два приложения с идентичным дизайном, осуществляющие загрузку, сортировку и сохранение массивов целых чисел. Эксперимент проводился на наборах данных разного объёма (1000, 10 000, 100 000 элементов). Время выполнение каждой операции измерялось отдельно. Для измерения времени в C++ использовалась стандартная библиотека `ctime`. Она содержит тип `clock_t` и функцию `clock`. Вместе они позволяют получить количество тактов процессора с момента запуска системы. Чтобы измерить длительность события (например, загрузки данных), вычитали начальное время из конечного. Для этого фиксировали время до и после события. Для Java использовался стандартный пакет `System`, в котором есть встроенная функция `currentTimeMillis`. Для измерения времени в миллисекундах использовалась функция `currentTimeMillis`. Время фиксировалось до и после события, затем вычислялась разница. У Java нет таких же возможностей для ручной оптимизации кода, как у C++, поскольку она компилируется собственным компилятором `javac` в байт-код, который затем выполняется на виртуальной машине Java (JVM). Этот компилятор автоматически оптимизирует код. Чтобы сделать сравнение максимально справедливым, среда разработки C++ была настроена на максимальную оптимизацию кода с использованием флага `O2` и запуск в режиме `Release`, что исключает накладные расходы на отладку и оптимизирует код настолько, насколько это возможно. Исходя из этого составим схему принципа работы системы (рис. 1).

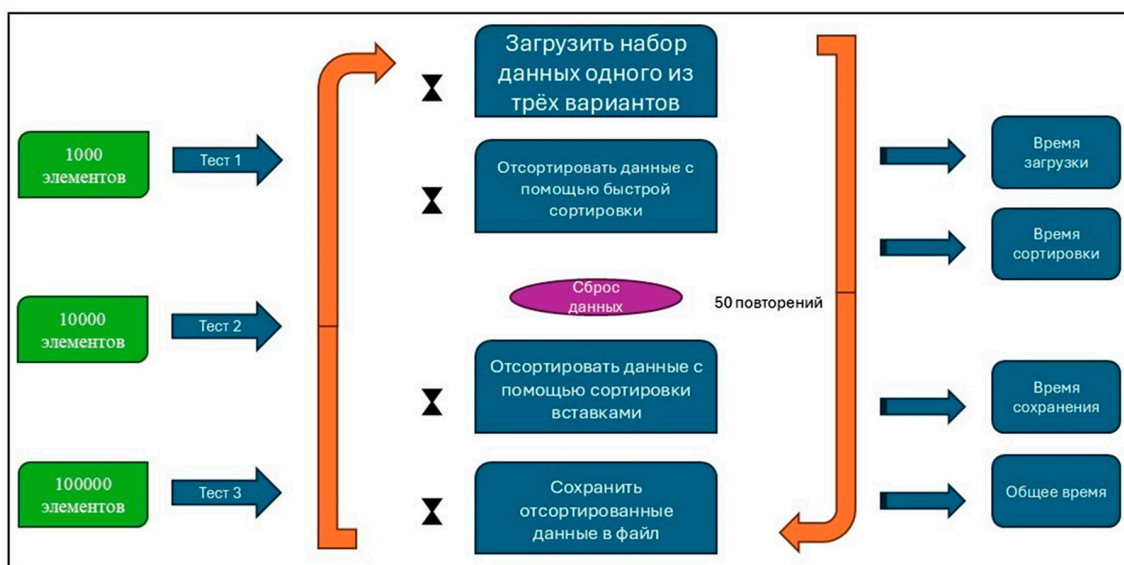


Рис. 1. Схема принципа работы

Каждый тест повторялся 50 раз для повышения достоверности результатов. Сравнительный анализ проводился на основе усреднённых значений времени выполнения. Приложения были созданы в разных интегрированных средах разработки, Java приложение было создано в VS Code, а приложение на C++ было запрограммировано в Visual Studio 2022. Оба проекта были созданы как 32-разрядные. Этот тест проводился на 64-разрядной операционной системе с Windows 11, 16 ГБ оперативной памяти и процессором i5, работающим на частоте 4,4 ГГц.

Результаты исследования и их обсуждение

Результаты показывают, что Java в целом работает быстрее, чем C++, а также быстрее загружает два из трёх наборов данных. C++ в целом быстрее при сортировке данных обоими алгоритмами и при сохранении в файлы. При чтении данных из файла посимвольно Java оказалась быстрее используя объекты File и Scanner, по сравнению с C++, который использовался стандартную библиотеку fstream. Java быстрее загрузила два больших набора данных, но была

медленнее при загрузке самого маленького – примерно на две секунды. Разница для других двух наборов составила около одной секунды для 10 000 целых чисел и около 22 секунд для 100 000. Медленная загрузка данных в C++ привела к тому, что общее время выполнения оказалась больше. В среднем, Java была быстрее благодаря более эффективному чтению файлов, однако при сортировке и обработке данных C++ в целом быстрее, особенно при сортировке самого большого набора с помощью сортировки вставками (на восемь секунд быстрее) и быстрой сортировки (на более чем секунду быстрее). Сохранение данных в конце каждой итерации было быстрее в C++ на несколько секунд, особенно для самого большого набора данных, где разница составила около двух секунд. Результаты показывают, что с увеличением размера наборов данных разница во времени растёт значительно. Масштабируемость Java лучше при загрузке данных, но при сортировке с помощью quicksort и сохранении данных C++ масштабируется лучше. Сортировка вставками плохо масштабируется в обоих приложениях. Таблицы и графики наглядно показывают эти различия (табл. 1, 2).

Таблица 1

Результаты тестирования производительности C++
(время выполнения в миллисекундах)

Размер набора данных	Load	Qsort	iSort	Save	Total
1000	0,9 мс	0,04 мс	0,1 мс	1,64 мс	2,68 ms
10000	7,5 мс	0,5 мс	12,62 мс	8,52 мс	29,14 ms
100000	73,18 мс	6,04 мс	1208,58 мс	16,94 мс	1304, 74 ms

Таблица 2

Результаты тестирования производительности Java
(время выполнения в миллисекундах)

Размер набора данных	Load	qSort	iSort	Save	Total
1000	2,34 мс	0,06 мс	0,22 мс	2,18 мс	4,8 мс
10000	6,44 мс	0,76 мс	12,48 мс	8,08 мс	27,76 мс
100000	51,28 мс	7,66 мс	1216,06 мс	19,9 мс	1249,9 мс

Ось Y графиков отображает размер набора данных, а ось X – время в миллисекундах. На первом графике отчетливо видно, что C++ (оранжевый) значительно медленнее и хуже масштабируется по сравнению с Java (синий)

при загрузке из файла. C++ быстрее только при работе с наименьшим набором данных. Так же нужно обратить внимание, что загрузка самого большого файла заняла больше семи секунд (73 миллисекунд) (рис. 2).

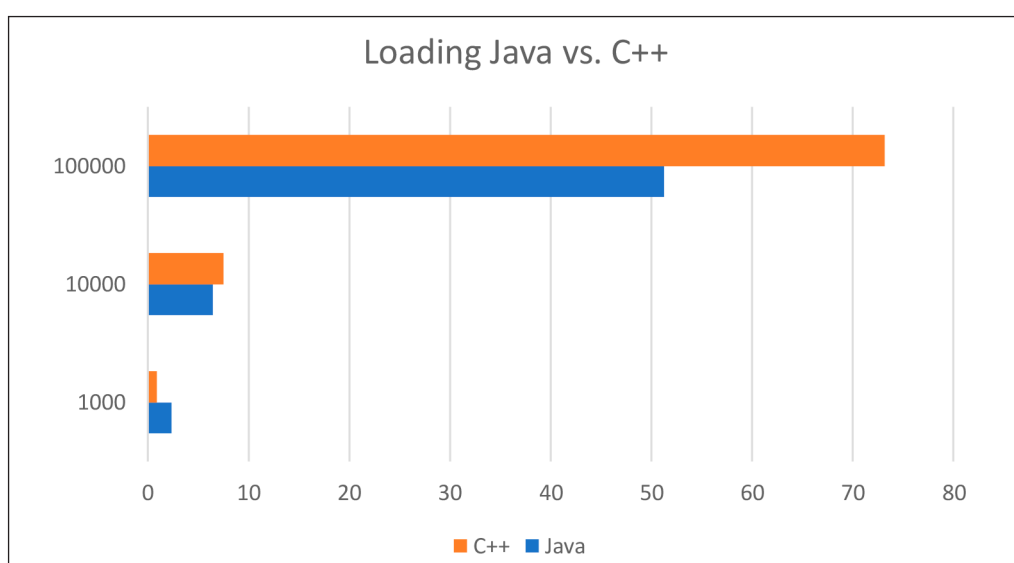


Рис. 2. Сравнение загрузки

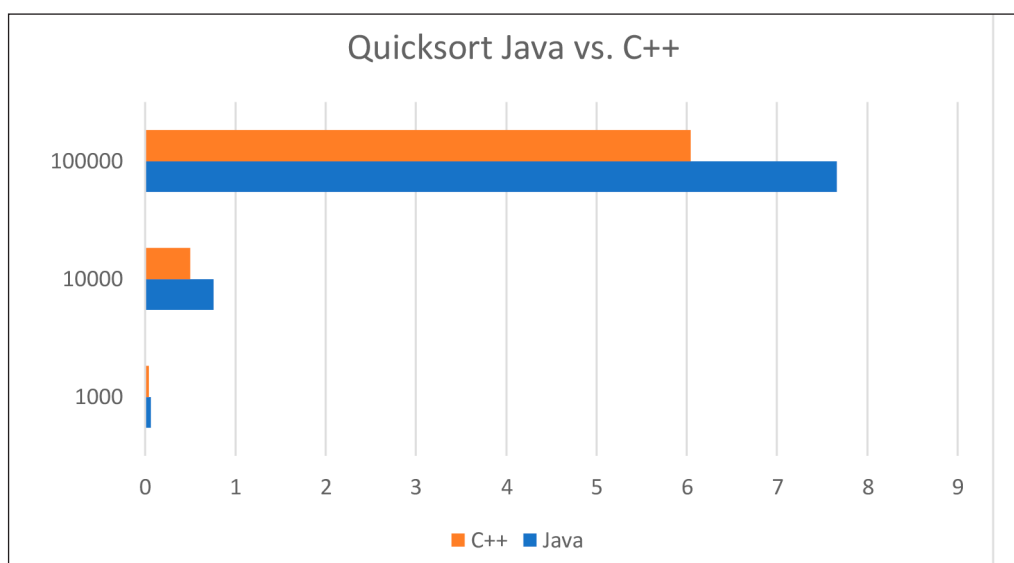


Рис. 3. Quicksort сравнение

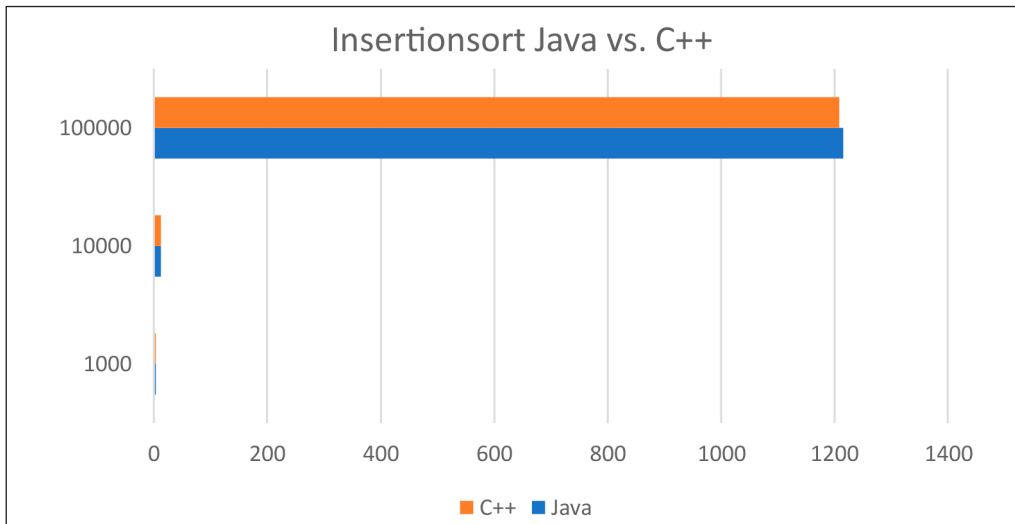


Рис. 4. Сравнение вставок и сортировки

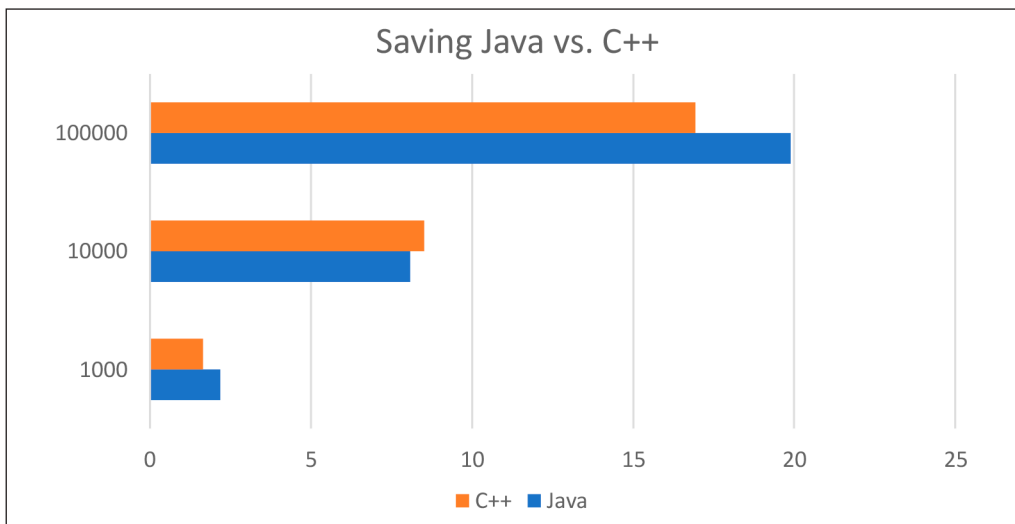


Рис. 5. Сравнение сохранения

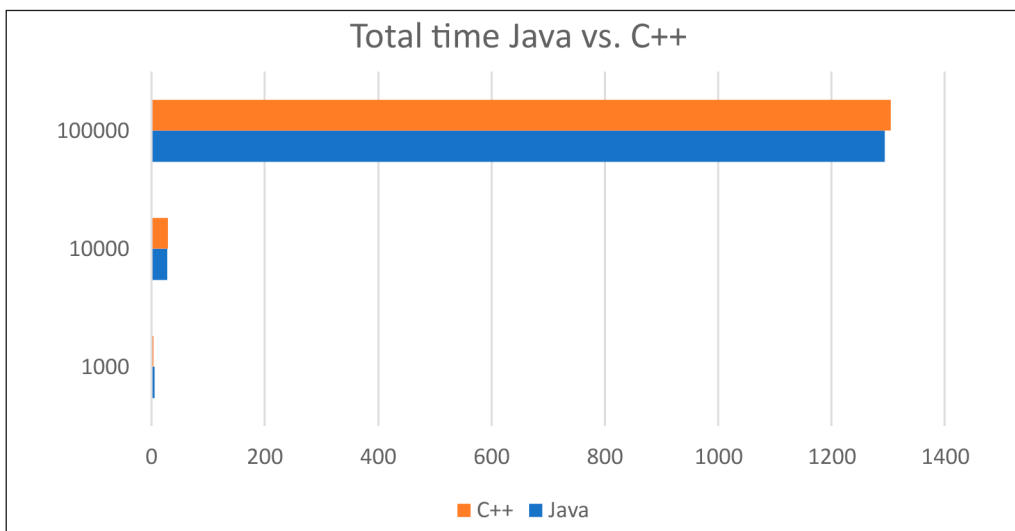


Рис. 6. Общее сравнение

На втором графике показаны результаты сравнения быстрой сортировки: C++ (оранжевый) оказалась быстрее, разница составляет всего несколько миллисекунд (рис. 3).

На третьем графике разница во времени не так велика, однако операция заняла почти 12 секунд (1200 мс). Алгоритм сортировки вставками плохо масштабируется и показала очень низкую производительность как в Java, так и в C++. C++ (оранжевый) справилась быстрее, но всего на несколько миллисекунд. Результаты для самого маленького набора данных на этом графике не видны, но C++ выполнила операцию за 0,1 мс, а Java – за 0,22 мс (рис. 4).

На четвёртом графике показано, что C++ был быстрее Java при сохранении данных в файл. Java был быстрее во втором тесте (рис. 5).

На пятом графике показано среднее общее время, затраченное приложениями на выполнение всех 50 итераций, Java (синий) оказалась медленнее C++ во всех тестах кроме одного – с самым маленьким набором данных (1000 целых чисел). Результаты для самого маленького набора трудно разобрать на этом графике, но общее время выполнения составило: C++ – 2,68 мс, Java – 4,8 мс (рис. 6).

Заключение

Результаты показывают, что C++ быстрее при сортировке и обработке данных, особенно при использовании quicksort и сортировки вставками, а также при сохранении файлов. Однако загрузка данных в C++ оказалась медленнее из-за посимвольного чтения, что привело к худшему масштабированию. В Java загрузка данных была быстрее благодаря более эффективной работе с токенами в библиотеке Scanner. В целом, C++ показала лучшую производительность при обработке данных, но Java при загрузке. Масштабируемость сортировки вставками оказалась плохой для обоих языков, в то время как quicksort масштабировался лучше. В реальных приложениях, где загрузка данных происходит не так часто, C++ может быть предпочтительнее благодаря более высокой скорости обработки.

Список литературы

1. Prechelt L. Technical opinion: Comparing java vs. C/C++ efficiency differences to interpersonal differences // Communications of the ACM. 1999. № 42(10). P. 109-112. DOI: 10.1145/317665.317683.
2. Nikishkov G. P., Nikishkov Y. G., Savchenko V. V. Comparison of C and java performance in finite element computations // Computers and Structures. 2003. № 81(24). P. 2401-2408. DOI: 10.1016/S0045-7949(03)00301-8.
3. METANIT: материалы по таким направлениям, как язык C#, C/C++ и технологии на базе Java. URL: <https://metanit.com/> (дата обращения: 15.12.2025).
4. Java currentTimeMillis function. (n.d.). Retrieved from. URL: [https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#currentTimeMillis\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/System.html#currentTimeMillis()) (дата обращения: 15.12.2025).
5. Шилдт Г. Java: руководство для начинающих. 7-е изд. М.: ООО «И.Д. Вильямс», 2020. 688 с.
6. Савченко В. В. Объектно-ориентированное программирование: учебное пособие. 2-е изд., перераб. и доп. М.: Академия, 2013. 288 с.

АРХИТЕКТУРА МАСШТАБИРУЕМОЙ ЛЕНТЫ НОВОСТЕЙ

Романенко В. В., Луцкий Е. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: romanenkoviktor080@yandex.ru

Научный руководитель: Луцкий Е. А.

Введение

В эпоху цифровых технологий социальные сети и информационные платформы накапливают колоссальные объёмы данных. По оценкам аналитиков, ежедневно в глобальной сети создаётся более 2,5 квинтиллиона байт информации, значительная часть которой поступает через приложения социальных сетей. Центральным компонентом таких платформ является лента новостей – динамически обновляемый набор контента, упорядоченный в соответствии с интересами каждого пользователя.

Традиционные подходы к организации информационных систем, основанные на централизованной архитектуре и реляционных базах данных, оказываются неэффективными при работе с такими объёмами данных. Необходимость обеспечить низкую задержку, высокую пропускную способность и одновременно сохранить логическую консистентность данных диктует новый подход к проектированию архитектуры. Существующие системы должны обрабатывать информацию в реальном времени, выстраивая персонализированные ленты для миллионов одновременно активных пользователей.

Задача построения масштабируемой ленты новостей объединяет несколько сложных проблем: управление потоками данных высокой пропускной способности, распределённое хранение информации, алгоритмическое ранжирование контента и кэширование часто запрашиваемых данных. Решение этих проблем требует применения современных технологий распределённых систем, включая системы обработки потоков данных, NoSQL базы данных и механизмы горизонтального масштабирования.

Цель данной статьи – изложить комплексный подход к проектированию архитектуры масштабируемой ленты новостей, рассмотрев все ключевые компоненты, их взаимодействие, и современные технические решения, применяемые крупными платформами.

Слой сбора и потоковой обработки данных

Первым критически важным компонентом архитектуры является система для сбора и обработки потоков данных в реальном времени. Apache Kafka выступает в качестве центрального брокера сообщений, обеспечивающего асинхронную доставку событий (публикация статей, лайки, комментарии и т.д.) от производителей данных к потребителям [1].

Архитектура Kafka основана на концепции распределённого, отказоустойчивого журнала событий. Ключевые преимущества [2]:

- Высокая пропускная способность: система способна обрабатывать сотни тысяч или миллионы событий в секунду;
- Надёжность доставки: репликация сообщений на нескольких узлах обеспечивает сохранение данных даже при отказе отдельных серверов;
- Масштабируемость: добавление новых партиций позволяет горизонтально расширять пропускную способность;
- Отстранение производителей и потребителей: асинхронная архитектура позволяет отделить логику создания контента от логики его обработки и доставки.

Для обработки потоков данных используются специализированные фреймворки, такие как Apache Flink, Apache Spark Streaming, Kafka Streams. Эти системы позволяют применять трансформации, фильтрацию и агрегацию к потокам событий в реальном времени. Например, система может немедленно агрегировать статистику просмотров, лайков и комментариев, и обновлять ранжирование контента на основе свежих данных [3].

Архитектура ленты новостей сочетает два подхода: потоковую обработку для немедленного обновления метрик и батч-обработку для более сложных аналитических задач и переобучения моделей машинного обучения. Такой гибридный подход, часто называемый Lambda архитектурой, позволяет достичь оптимального баланса между скоростью и точностью обработки.

При проектировании сервиса формирования персональной ленты ключевым архитектурным решением является выбор стратегии распространения новых публикаций и событий. На практике выделяют три подхода [4]:

- Fan-out on write. При записи нового поста система проактивно распределяет запись в пользовательские ленты. Это позволяет получить очень быстрые операции чтения, однако записи становятся затратными – при наличии «звёздных» аккаунтов количество операций записи растёт линейно с числом подписчиков и может стать узким местом;
- Fan-out on read (pull). При обращении пользователя система динамически генерирует ленту, агрегация кандидатов и ранжирование выполняются на чтении. Записи дешёвые, но время ответа и нагрузка на службе чтения выше;
- Гибридный подход. Комбинация: precompute для большинства «обычных» подписчиков и on-read для «звёздных». Также часто precompute выполняется лишь для теплой части аудитории или для «звёздных» сигналов. Такой подход даёт компромисс между стоимостью записи и целевыми SLA на чтение.

Распределённое хранилище данных

Для хранения масштабируемой ленты новостей требуются базы данных, способные эффективно работать с неструктурированными или слабоструктурированными данными и предусматривающие горизонтальное масштабирование. NoSQL – стандартный выбор в этой области [5].

Основные NoSQL технологии:

- Apache Cassandra – распределённая СУБД с ориентиром на высокую доступность и большую пропускную способность при записи. Использует модель широких столбцов и проектируется под конечную (eventual) согласованность данных;
- MongoDB – документоориентированная база, хранящая записи в виде JSON-подобных документов. Характеризуется гибкой схемой данных и поддержкой транзакций на уровне отдельных документов;
- Redis – in-memory хранилище «ключ-значение», обеспечивающее чрезвычайно низкую задержку доступа. Широко применяется для кэширования, управления сессиями и хранения горячих данных.

Одной из проблем масштабирования является распределение данных между несколькими серверами. Шардирование – техника, при которой набор данных разбивается на логические подмножества, каждое из которых хранится на отдельном узле.

Стратегии шардирования:

- Шардирование по пользователю: все данные одного пользователя хранятся на одном шарде, что обеспечивает быстрый доступ при чтении ленты;
- Шардирование по идентификатору поста: данные о каждом посте распределяются по шардам, что облегчает репликацию и параллельную обработку;
- Адаптивное шардирование: система динамически перераспределяет данные в зависимости от текущей нагрузки, перемещая часто используемые данные на узлы с большей пропускной способностью.

Кэширование и ускорение доступа

Кэширование является критическим компонентом для достижения низкой задержки. Архитектура включает несколько уровней кэша [6]:

- Кэш на уровне приложения: данные, часто запрашиваемые приложением, хранятся в оперативной памяти приложения;
- Кэш на уровне сервиса: общий кэш Redis, разделяемый между несколькими экземплярами приложения, обеспечивает согласованность данных и снижает нагрузку на основное хранилище;
- Кэш на уровне CDN: контент кэшируется на географически распределённых узлах Content Delivery Network для ускорения доставки конечным пользователям.

Поскольку объём кэша ограничен, необходимо определить, какие данные удалять при переполнении. Наиболее распространённые стратегии [6]:

- LRU (Least Recently Used): удаляются данные, к которым давно не было обращений;
- LFU (Least Frequently Used): удаляются данные с наименьшим числом обращений;
- TTL (Time To Live): данные автоматически удаляются по истечении заданного времени жизни.

Инвалидация кэша требует синхронизации между системой хранения и кэшем. При обновлении данных в основной БД соответствующие записи в кэше должны быть немедленно обновлены или удалены.

Алгоритмы ранжирования контента

Ядром ленты новостей является алгоритм, определяющий порядок отображения контента. Эффективное ранжирование учитывает множество факторов:

- Социальный граф: посты от близких друзей обычно имеют более высокий приоритет;
- История взаимодействия: система анализирует, какой контент пользователь обычно лайкает или комментирует, и предпочитает подобный контент;
- Временной компонент: свежий контент обычно более интересен, чем старый;
- Метаданные контента: тематика, жанр, язык поста;
- Ранжирование может быть реализовано на основе различных моделей машинного обучения, от простых линейных моделей до глубоких нейронных сетей.

Основным подходом является коллаборативная фильтрация – метод, предполагающий, что пользователи с похожими интересами будут одинаково оценивать один и тот же контент.

Методы коллаборативной фильтрации:

- На основе пользователей: находят пользователей с похожими предпочтениями и рекомендуют контент, понравившийся соседним пользователям;
- На основе элементов: если пользователю понравился пост, рекомендуются похожие посты на основе выученных векторных представлений – embeddings;
- Матричная факторизация: предпочтения пользователей разлагаются в виде произведения двух матриц меньшего ранга, что позволяет выявить скрытые признаки.

Первым этапом для рекомендательной системы обычно является генерация ограниченного набора кандидатов из огромного пула контента. Современные практики используют векторные представления элементов и пользователей и применяют методы приближённого поиска ближайших соседей (ANN) для масштабируемого извлечения похожих элементов [7].

Интеграция в pipeline:

- комбинировать несколько генераторов (graph-based, content-based via ANN, recent windows и business-rules);
- использовать ANN-индексы для предварительного отбора ~100 – 1000 кандидатов, далее применять быстрый LTR и затем более тяжёлые re-rankers;
- учитывать требования к задержке при выборе реализации ANN.

Обеспечение надёжности и консистентности

Для обеспечения отказоустойчивости каждая единица данных реплицируется на несколько узлов. Стратегии репликации:

- Синхронная репликация: изменение считается завершённым только после того, как оно записано на все реплики. Обеспечивает строгую консистентность, но замедляет операции;
- Асинхронная репликация: клиент получает подтверждение после записи на основной узел, а репликация на другие узлы происходит позднее. Быстрее, но несёт риск потери данных при сбое основного узла.

В распределённых системах возникают сложности при обработке конфликтов между записями, выполняемыми на разных узлах. Разрешение конфликтов может быть основано на временных метках или на применении пользовательских функций слияния [8].

Мониторинг и оптимизация производительности

Операционная сложность масштабируемой ленты новостей требует постоянного мониторинга ключевых метрик производительности:

- Задержка: время отклика системы на запрос;
- Пропускная способность: количество обработанных запросов в единицу времени;
- Доступность: процент времени, в течение которого система функционирует без ошибок.

Инструменты для мониторинга и логирования такие как: ELK Stack, Prometheus, Grafana позволяют выявлять узкие места и оптимизировать системные параметры.

Заключение

Проведённое исследование позволило рассмотреть проблему проектирования масштабируемой ленты новостей и сформировать архитектурное решение, ориентированное на обработку больших объёмов данных в режиме, близком к реальному времени. В ходе работы была разработана архитектура, объединяющая системы потоковой обработки событий, распределённые NoSQL-хранилища, механизмы горизонтального масштабирования и многоуровневого кэширования. Использование брокера сообщений и стриминговых фреймворков обеспечивает высокую пропускную способность и оперативное об-

новление метрик взаимодействия пользователей с контентом. Применение стратегий шардирования и репликации позволяет достичь отказоустойчивости и гибкости системы при росте нагрузки, а использование кэширования существенно снижает задержку при формировании ленты. Рассмотренные алгоритмы ранжирования и рекомендательные подходы обеспечивают персонализацию контента с учётом социальных связей, истории взаимодействий и временных факторов. Полученные результаты подтверждают, что предложенная архитектура соответствует требованиям современных информационных платформ и может быть использована в качестве практической основы для построения высоконагруженных систем формирования новостных лент.

Список литературы

1. Kreps J., Narkhede N., Rao J. Kafka: A distributed messaging system for log processing, 2011.
2. Singh I., Vivek C. Real-time event joining in practice with Kafka and Flink // arXiv preprint arXiv:2410.15533, 2024.
3. Carbone P., Katsifodimos A. et al. Apache Flink: stream and batch processing in a single engine // Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2015.
4. Алексеева К. А. Применение паттерна FAN-OUT для обновления лент в высоконагруженных веб-приложениях: выпускная квалификационная работа. Минск: БГУИР, 2025.
5. Davoudian A., Chen C., Liu M. A survey on NoSQL stores // ACM Computing Surveys (CSUR), 2018.
6. Megiddo N., Modha D.S. ARC: A self-tuning, low overhead replacement cache // Proceedings of the 2nd USENIX Symposium on File and Storage Technologies (FAST), 2003.
7. Ульянов М. В. Приближённые ближайшие соседи Москва, 2015.
8. Bailis P., Venkataraman S., Franklin M.J., et al. Probabilistically Bounded Staleness for Practical Partial Quorums, 2012.

ОБЗОР МЕТОДОВ ОЦЕНКИ ИНВЕСТИЦИЙ В КИБЕРБЕЗОПАСНОСТИ. ПРОЦЕСС УПРАВЛЕНИЯ РИСКАМИ

Романюк М. А., Мозговенко А.А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@mozgovenko.ru*

Задачи исследования:

1. Систематизировать основные виды угроз, связанных с распространением сообщений в социальных сетях.
2. Оценить масштаб и последствия угроз для отдельных пользователей и социума.

Материалы и методы исследования

Современные исследования в области цифровой безопасности выделяют несколько ключевых направлений анализа угроз в социальных сетях:

1. Киберпреступность и мошенничество. Работы последних лет (2022–2025 гг.) акцентируют внимание на эволюции фишинговых атак и методах социальной инженерии. Отмечается рост использования deepfake – технологий

для создания поддельных сообщений и имитации доверенных источников.

2. Дезинформация и манипулирование. Исследования в области медиапсихологии и политической коммуникации фиксируют усиление роли соцсетей как каналов распространения фейков и пропаганды. Особое внимание уделяется алгоритмическому усилению дезинформации и её влиянию на общественное мнение.

3. Психологические угрозы. Публикации по киберпсихологии описывают рост случаев кибербуллинга, stalkingа и онлайн харасмента. Анализируются долгосрочные последствия для психического здоровья жертв, включая тревожные расстройства и суицидальные мысли.

4. Утечка данных и приватность. Исследования в сфере информационной безопасности демонстрируют, что сообщения в соцсетях часто становятся источником компрометации персональных данных через:

- скрапинг информации;
- анализ метаданных;
- эксплуатацию уязвимостей платформ.

5. Экстремистские угрозы. Работы по криминологии и противодействию терроризму отмечают использование соцсетей для вербовки и координации противоправной деятельности. Активно изучаются методы выявления и блокировки экстремистского контента.

Основная цель – создать типологию угроз, порождаемых сообщениями в социальных сетях.

Результаты исследования и их обсуждение

С увеличением использования социальных сетей многие пользователи стали уязвимыми к угрозам своей приватности и безопасности. Эти опасности могут быть разделены на 4 главные категории (рис. 1.1). Первая категория содержит классические угрозы, в частности, угрозы конфиденциальности и безопасности, которые не только угрожают пользователям соцсетей, но и пользователям Интернета, не использующим социальные сети. Вторая категория охватывает современные угрозы, то есть угрозы, в основном уникальные для среды социальных сетей, и которые используют инфраструктуру социальных сетей для угрозы конфиденциальности и безопасности пользователя. Третья категория состоит из комбинированных угроз, где мы описываем, как сегодняшние нападающие могут, и часто делают, совмещать разные типы атак для создания более сложных и летальных приступов. Четвертая и последняя категория включает в себя угрозы, специально ориентированные на детей, использующих социальные сети.

Классические угрозы, такие как вредоносное ПО, фишинг, спам и межсайтовый скриптинг (XSS), остаются актуальными в эпоху социальных сетей. Они быстро распространяются среди пользователей, используя их личные данные и доверие.

Вредоносное ПО, как Koobface, заражает компьютеры пользователей социальных сетей и создает бот-сети. Фишинговые атаки, например в Facebook (соцсеть признана экстремистской и запрещенной на территории России, заблокирована РКН), обманывают пользователей, заставляя их раскрывать конфиденциальную информацию. Спамеры используют фальшивые профили для распространения рекламы. XSS-атаки, такие как червь Mikeuu, распространяются через уязвимости в соцсетях.

Интернет-мошенники, используя соцсети, получают доверие жертв и крадут их данные. Например, в 2010 году мошенники взломали аккаунт Эбигейл Пикетт в Facebook (соцсеть признана экстремистской и запрещенной на территории России, заблокирована РКН), чтобы выманить деньги у ее друзей.

Современные угрозы часто нацелены на личную информацию пользователей и их друзей. Злоумышленники могут создавать поддельные профили, собирать данные через друзей или использовать атаки вывода для получения конфиденциальных сведений.

Примеры угроз

1. Clickjacking – обман, заставляющий пользователей нажимать не то, что они хотели.

2. Деанонимизация – раскрытие настоящей личности через файлы cookie, группы и топологию сети.

3. Распознавание лица – создание биометрической базы данных из публичных фото.

4. Поддельные профили – автоматизированные аккаунты для сбора личных данных.

5. Атаки клонирования личности – дублирование профиля для обмана друзей и сбора информации.

6. Атаки вывода – анализ общедоступных данных для получения скрытой информации.

7. Утечка локации – обмен данными о местонахождении через соцсети.

8. Socware – вредоносные сообщения и приложения, маскирующиеся под друзей.

Дети, как малые, так и подростки, подвергаются угрозам в социальных сетях, включая онлайн-хищников, рискованное поведение и кибербуллинг.

Онлайн-хищники: Педофилы в интернете могут использовать детей для производства и распространения детской порнографии, а также для онлайн- или офлайн-эксплуатации. Исследования показывают, что контент, контакты с взрослыми и рискованное поведение детей могут привести к серьезным последствиям.

Рисковое поведение: Прямое общение с незнакомцами, интимные разговоры, предоставление личной информации и фотографий – все это увеличивает риск для детей. Комбинация этих действий может быть особенно опасной.

Кибербуллинг: Издевательства в интернете через электронные письма, чаты и соцсе-

ти, включающие публикацию оскорбительных материалов и угроз, особенно сильно влияют на детей. Опрос показал, что 12% родителей сталкивались с кибербуллингом своих детей, чаще всего на платформах вроде Facebook (соцсеть признана экстремистской и запрещенной на территории России, заблокирована РКН).

Радикализация: Террористические организации, такие как ИГИЛ, используют соцсети для вербовки и распространения информации о своих планах. Борьба с такими угрозами требует обнаружения и пресечения распространения вредоносных сообщений.

Заключение

Сообщения в социальных сетях выступают катализатором многообразных угроз: от индивидуального мошенничества до масштабных информационных кампаний. Их объединяет высокий уровень маскировки под легитимный контент.

Ключевые угрозы:

- фишинг и социальная инженерия (поддельные ссылки, имитация доверенных лиц);
- дезинформация (фейки, манипулятивные нарративы);
- кибербуллинг (травля, доксинг);
- утечка данных (скрапинг, анализ цепочек сообщений);
- экстремистская агитация (вербовка, координация незаконных действий).

Список литературы

1. Kotenko I., Vitkova L., Saenko I., Tushkanova O., Branitskiy A. The intelligent system for detection and counteraction of malicious and inappropriate information on the Internet // AI Communication. 2020. Vol. 33. № 1. P. 13-25. DOI: 10.3233/aic-200647.
2. Vitkova L., Kotenko I., Kolomeets M., Tushkanova O., Chechulin A. Hybrid Approach for Bots Detection in Social Networks Based on Topological, Textual and Statistical Features // Proceedings of the Fourth International Scientific Conference Intelligent Information Technologies for Industry. Springer. 2020. Vol. 1156. P. 412-421. DOI: 10.1007/978-3-030-50097-9_42.
3. Liu L., Peng, T. Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF // Journal of Information Science and Engineering. 2014. Vol. 30. P. 1463-1481.
4. Li X.L., Liu B., Ng S. K. Negative training data can be harmful to text classification // Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics. 2010. P. 218-228.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИДЕНТИФИКАЦИИ ОБЪЕКТОВ ВОЗДУШНОГО ПРОСТРАНСТВА

Рукас М. К., Букреев Д. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru

Научный руководитель: Букреев Д. А.

Введение

Современное воздушное пространство характеризуется высокой динамичностью, ростом плотности воздушного движения и быстрым

распространением беспилотных летательных аппаратов, используемых как в гражданских, так и в потенциально опасных целях. Увеличение числа малозаметных, манёвренных и разнородных объектов существенно усложняет задачи их наблюдения, сопровождения и идентификации. Традиционные методы радиолокационного и оптико-электронного контроля, основанные преимущественно на ручной классификации или пороговой обработке сигналов, оказываются недостаточно эффективными при высокой скорости изменения обстановки и большом объёме информации. Это приводит к росту нагрузки на оператора и повышает риск возникновения ошибок, особенно в условиях помех, недостаточной видимости и неоднородности данных. В этих условиях становится актуальным переход к интеллектуальным автоматизированным системам анализа, использующим методы искусственного интеллекта и машинного обучения. Нейронные сети обладают способностью выявлять сложные взаимосвязи в данных и адаптироваться к изменяющимся условиям наблюдения, что делает их перспективным инструментом для решения задач идентификации воздушных объектов. Применение глубоких сверточных архитектур, детекторов реального времени и гибридных нейросетевых моделей позволяет повысить точность классификации, сократить задержку обработки видеопотоков и снизить влияние человеческого фактора. Такие технологии создают предпосылки для построения комплексных систем мониторинга, способных автоматически распознавать объекты различных типов, прогнозировать их поведение и предоставлять оператору структурированную информацию для принятия решений. Предлагаемый подход обеспечивает адаптацию модели к изменяющимся условиям наблюдения и повышение достоверности идентификации при ограниченных вычислительных ресурсах. Практическая значимость заключается в возможности применения разработанной системы в комплексах контроля воздушного пространства, центрах управления БПЛА, системах видеонаблюдения и учебно-тренировочных платформах.

Цель исследования – автоматизация процесса идентификации объектов воздушного пространства, путем использования методов искусственных нейронных сетей и ориентированной на функционирование в режиме реального времени.

Материалы и методы исследования

Исследование автоматизированной системы идентификации объектов воздушного пространства опирается на совокупность методов анализа данных, машинного обучения и технологий компьютерного зрения, обеспечивающих интеллектуальную обработку визуальной информации в реальном времени. Методологическая

база включает изучение современных подходов к классификации и распознаванию объектов, анализ архитектур глубоких нейронных сетей, а также разработку алгоритмов, обеспечивающих стабильность функционирования системы при изменяющихся условиях наблюдения.

Материалом исследования служат изображения и видеопотоки, полученные от оптических и инфракрасных средств наблюдения. Данные характеризуются высокой вариативностью – различием ракурсов, расстояний, освещённости, погодных условий и степенью контрастности изображений. Для подготовки обучающих выборок используются специализированные наборы данных, включающие изображения самолётов, вертолётов, беспилотных аппаратов, птиц и неидентифицируемых объектов. Для повышения репрезентативности выборки применяется аугментация данных: масштабирование, повороты, отражения, варьирование яркости и контраста. Такой подход позволяет повысить устойчивость модели к внешним искажениям и расширить диапазон условий, при которых сохраняется корректность идентификации.

Результаты исследования и их обсуждение

Архитектура автоматизированной системы идентификации объектов воздушного пространства представляет собой многоуровневую модульную структуру, предназначенную для комплексной обработки визуальных данных в реальном времени. Основная идея заключается в объединении разнородных сенсорных источников, интеллектуальных нейросетевых алгоритмов и управляющих компонентов в единую вычислительную платформу, обеспечивающую высокую скорость анализа, устойчивость к внешним воздействиям и возможность дальнейшего масштабирования [1]. Формирование архитектуры осуществлялось на основе анализа современных систем ситуационной осведомлённости и опыта внедрения нейронных сетей в задачи мониторинга воздушного пространства.

1. Принципы построения архитектуры

При проектировании архитектуры были приняты ключевые системообразующие принципы:

Модульность – каждый компонент отвечает за строго определённую функцию, что обеспечивает взаимозаменяемость и возможность модернизации отдельных модулей без изменения всей системы. Модульная структура позволяет использовать различные модели детекции, алгоритмы классификации или форматы данных, адаптируя подсистему под конкретные условия наблюдения.

Потоковая обработка данных – система функционирует в режиме непрерывного анализа видеопотока, что требует асинхронного выполнения задач: предобработка кадров, детекция,

классификация и фильтрация происходят параллельно. Такое решение минимизирует задержки и обеспечивает работу в реальном времени – одно из ключевых требований для задач контроля воздушного пространства.

Параллелизм вычислений – использование графических ускорителей (GPU) и оптимизированных библиотек глубокого обучения позволяет выполнять детекцию и классификацию одновременно для нескольких объектов и кадров. Это особенно важно при высокой плотности воздушной обстановки, когда в поле зрения находится несколько целей.

Адаптивность – система должна корректно работать при изменении условий освещённости, появлении шумов, изменении ракурса наблюдения и параметров видеопотока. Для этого в архитектуре предусмотрены механизмы дообучения модели, корректировки параметров фильтрации и обновления конвейера обработки данных.

Интегрируемость – архитектура предусматривает возможность подключения к существующим системам радиотехнического и оптико-электронного наблюдения, что расширяет потенциальные области применения – от гражданского контроля до задач ПВО.

Все принципы реализованы в общей структурной схеме, включающей пять функциональных подсистем:

1) Подсистема сбора и первичной обработки данных – отвечает за получение изображений и видеопотоков от оптических, инфракрасных и при необходимости радиолокационных средств наблюдения [5]. На данном уровне выполняются:

- захват и синхронизация кадров;
- нормализация форматов изображения (RGB → Grayscale);
- пространственная стабилизация и подавление шумов;
- коррекция яркости и контраста при отсутствии стабильных условий наблюдения.

2) Подсистема детекции объектов – выделение областей интереса (ROI) в каждом кадре. В системе применяется одношаговая архитектура YOLO (You Only Look Once), позволяющая анализировать изображение целиком, без предварительного выделения регионов вручную.

Преимущества использования YOLO:

- высокая скорость обработки (до 50–60 FPS в оптимизированных конфигурациях);
- устойчивость при работе с объектами различных размеров;
- способность корректно распознавать динамичные цели.

Модель выдаёт координаты рамок и вероятности наличия объектов. Эти данные затем передаются в классификатор, что формирует связанный алгоритмический поток. YOLO выступает первым интеллектуальным фильтром

и существенно снижает нагрузку на дальнейшие этапы обработки.

3) Подсистема идентификации и классификации объектов – является ядром всей архитектуры и содержит глубокую сверточную нейросеть, выполняющую непосредственную идентификацию воздушных объектов [6]. В качестве основной архитектуры используется модификация ResNet-50 и EfficientNet, выбранные благодаря:

- высокой точности классификации;
- устойчивости к искажениям изображения;
- умеренным вычислительным требованиям;
- способности к дообучению и fine-tuning.

На вход классификатора поступают вырезанные фрагменты (ROI), выделенные подсистемой детекции. Нейросеть формирует вероятностное распределение классов: «самолёт», «вертолёт», «БПЛА», «птица», «неопознанный объект». При обработке сложных сцен может применяться ансамблевый подход – например, комбинация CNN и Vision Transformer. Это увеличивает надёжность идентификации при низкой контрастности или частичном перекрытии цели [3].

4) Подсистема принятия решения и фильтрации – результаты нейросетевой классификации проходят дополнительную проверку для исключения ложных срабатываний. Механизм принятия решения включает:

- пороговую фильтрацию по вероятности;
- анализ динамики движения объекта;
- временную корреляцию последовательности кадров;
- проверку согласованности поведения цели.

Для временной коррекции используется рекуррентная нейросеть, анализирующая изменение положения объекта в соседних кадрах. Это особенно важно, когда объект частично перекрыт, быстро меняет траекторию или присутствуют атмосферные артефакты. Модуль принятия решения формирует окончательную метку класса и передаёт данные в подсистему визуализации.

5) Подсистема визуализации, хранения и управления – обеспечивает пользовательский интерфейс и контроль работы всей системы. Основные функции:

- отображение видеопотока с выделенными объектами и вероятностями;
- ведение журнала событий с временными метками и координатами;
- сохранение результатов в базе данных;
- передача данных в внешние системы мониторинга;
- режим ручной верификации оператором.

Хранение данных обеспечивает возможность анализа эффективности, построения статистических моделей и последующего дообучения нейросетей, что является важным элементом адаптивной архитектуры.

Модульность архитектуры позволяет расширять функциональность без изменения ба-

зового контура системы, что делает решение технологически перспективным и готовым к интеграции в реальные комплексы мониторинга воздушного пространства.

2. Алгоритмическое обеспечение и нейросетевая модель

Алгоритмическое обеспечение разработанной системы идентификации формирует её интеллектуальное ядро и определяет характер взаимодействия всех функциональных модулей – от обработки исходных данных до формирования финального решения. В отличие от традиционных алгоритмов, основанных на пороговой фильтрации или ручном выделении признаков, предложенное решение опирается на принципы глубокого обучения, что позволяет значительно расширить возможности системы при работе в условиях неопределённости, помех и высокой динамики воздушной обстановки [4]. Такой подход обеспечивает переход от жёстко заданных правил к адаптивной модели поведения, способной корректировать свои решения в зависимости от текущих условий наблюдения.

Центральным элементом алгоритмической части является последовательный конвейер обработки данных. Он включает несколько взаимосвязанных этапов, каждый из которых вносит собственный вклад в повышение достоверности идентификации. Потокоевое функционирование позволяет системе работать практически без задержек, что особенно важно для видеонаблюдения, где информация быстро устаревает.

Первичная обработка видеопотока направлена на обеспечение стабильности входных данных. Изображения, получаемые от камеры или тепловизора, могут отличаться по яркости, резкости, наличию шумов, а при работе на открытой местности – также по степени воздействия атмосферных факторов. Поэтому на этом этапе выполняются операции нормализации, мягкого шумоподавления и коррекции контрастности. Они не только улучшают визуальные характеристики кадра, но и создают благоприятные условия для корректной работы нейросетевых моделей, чувствительных к нарушениям структуры изображения.

Для решения задачи детекции объектов применяется детектор YOLO, который эффективно совмещает высокую скорость обработки с устойчивой локализацией объектов различных размеров [2]. YOLO анализирует изображение целиком, не разделяя его на отдельные области вручную, что сокращает время обработки и повышает согласованность результатов при работе с динамичными сценами. Детектор формирует набор областей интереса, передаваемых далее на классификацию.

Процедура классификации является центральным моментом всей системы, поскольку именно здесь принимается решение о том,

к какому типу относится наблюдаемый объект [8]. Для решения этой задачи была выбрана глубокая сверточная нейронная сеть ResNet-50, отличающаяся высокой устойчивостью к искажениям и способностью выделять сложные пространственные признаки даже в условиях низкой контрастности или частичного перекрытия объекта. Наличие остаточных связей позволяет модели сохранять качество обучения при значительной глубине, а использование предобученных весов ускоряет адаптацию сети под специфику воздушных объектов. В ряде сценариев классификация дополняется использованием архитектур EfficientNet или Vision Transformer, что обеспечивает повышение устойчивости модели в сложных визуальных условиях.

Ключевым фактором успешного функционирования нейросетевой части является качество обучающей выборки. Для её формирования использовались изображения воздушных объектов различного типа, полученные при разных условиях освещения и на разных расстояниях. Чтобы повысить способность модели к обобщению, применялись методы аугментации – масштабирование, изменение яркости, повороты и добавление шумов. Это позволило имитировать реальные изменения условий наблюдения и обеспечить высокую устойчивость модели к непредвиденным ситуациям.

Финальный этап алгоритма связан с анализом и интерпретацией результатов классификации. Даже самые точные модели глубокого обучения склонны давать колебания на отдельных кадрах, особенно если объект движется быстро, частично перекрыт или находится в зоне низкой видимости [7]. Поэтому система использует дополнительный механизм фильтрации, основанный на проверке временной стабильности и согласованности движения. Если модель фиксирует объект в нескольких соседних кадрах с высокой вероятностью, система подтверждает идентификацию. В противном случае результат подвергается повторной проверке или передаётся на анализ оператору. Такой подход позволяет существенно снизить число ложных тревог, сохраняя при этом способность системы реагировать на реальные угрозы.

3. Тестирование и оценка эффективности функционирования системы

Оценка работоспособности разработанной системы является важнейшим этапом исследования, поскольку именно она позволяет определить, насколько успешно реализованные алгоритмы справляются с задачами детекции и идентификации в условиях, близких к реальной эксплуатации.

Тестирование проводилось на вычислительной платформе, оснащённой графическим ускорителем NVIDIA RTX 3060, что обеспечило необходимую вычислительную мощность

для работы глубоких нейросетевых моделей. В качестве входных данных использовались изображения и короткие видеосегменты, содержащие примеры различных объектов воздушного пространства – самолётов, вертолётов, беспилотных летательных аппаратов, а также птиц и атмосферных объектов, способных вызывать ложные срабатывания. Такая выборка позволила оценить способность системы различать визуально сходные объекты и обрабатывать сцены различной сложности. Всего в тестировании было задействовано около 6000 изображений и 800 видеоклипов, отражающих широкий диапазон условий наблюдения.

Особое внимание уделялось вариативности входных данных: они отличались по разрешению, уровню контрастности, освещённости и степени зашумлённости. Это позволило имитировать реальные условия эксплуатации, где наблюдение может вестись как в дневное, так и в сумеречное время, при наличии тумана, облачности или отражений от наземных объектов.

Одним из ключевых требований к системе является способность функционировать в реальном времени, поскольку задержка при идентификации даже в несколько секунд может привести к потере актуальности данных. Проведённые испытания показали, что система демонстрирует устойчивую скорость обработки видеопотока – в среднем 27–30 кадров в секунду при разрешении 1280×720. Этот показатель остаётся стабильным даже при наличии нескольких объектов в кадре, благодаря разумному распределению нагрузки между детектором и классификатором. Средняя задержка между поступлением кадра и выдачей результата не превышала 140 мс.

Помимо скорости обработки оценивалась устойчивость функционирования системы при изменении условий наблюдения. Даже при существенном ухудшении освещённости или появлении атмосферных искажений система продолжала корректно выделять объекты, а снижение точности оставалось в пределах 5–6 %. Это говорит о том, что выбранные методы предобработки и архитектуры нейронных сетей обладают достаточной адаптивностью.

Ключевыми метриками качества работы системы стали accuracy, precision, recall и F1-score, позволяющие комплексно оценить способность модели правильно классифицировать объекты и избегать ложных срабатываний. По результатам тестирования средняя точность классификации составила 94,7 %, что является высоким показателем, учитывая сложность входных данных и большое количество классов. Метрики precision и recall составили 93,1 % и 92,4 % соответственно, что свидетельствует о сбалансированности модели: она не только правильно идентифицирует большинство объектов, но и практически не пропускает реальные

цели. Система демонстрирует устойчивое поведение при анализе динамичных сцен: последовательная фильтрация на основе временной корреляции позволяет нивелировать случайные ошибки классификации, особенно при появлении быстро перемещающихся или частично перекрытых объектов. Уровень ложных срабатываний не превышал 4,1 %, что является приемлемым значением для задач мониторинга, где ошибочная классификация должна быть сведена к минимуму.

Заключение

Проведённое исследование позволило всесторонне рассмотреть проблему автоматизированной идентификации объектов воздушного пространства и разработать функционально завершённую систему, способную решать задачи мониторинга в условиях реальной эксплуатации. В ходе исследования была сформирована архитектура системы, объединяющая принципы модульности, потоковой обработки данных и адаптивности. Такое построение обеспечивает её гибкость, возможность расширения и интеграции с существующими средствами наблюдения. Нейросетевые модули, основанные на современных архитектурах глубокого обучения, продемонстрировали высокую эффективность при обработке визуальной информации, позволяя уверенно выделять и классифицировать воздушные объекты различной природы. Особое внимание было уделено разработке алгоритмического обеспечения. Использование сочетания детектора YOLO и классификационных моделей типа ResNet обеспечило устойчивое распознавание объектов даже в условиях недостаточной освещённости, шумов или частичного перекрытия цели. Система продемонстрировала способность функционировать в режиме реального времени, а реализованные методы временной фильтрации позволили снизить количество ложных срабатываний и повысить обоснованность принимаемых решений. Проведённые испытания подтвердили работоспособность и надёжность системы. Средняя частота обработки видеопотока достигала 27–30 кадров в секунду, что соответствует требованиям к оперативным системам наблюдения. Точность классификации превысила 94 %, а устойчивость к изменениям внешних условий показала, что системы на основе глубокого обучения могут эффективно использоваться для задач воздушного мониторинга. Сравнение с существующими решениями показало явные преимущества разработанной системы по точности, скорости обработки и уровню ложных срабатываний. Таким образом, разработанная автоматизированная система идентификации объектов воздушного пространства сочетает в себе современные методы искусственного интеллекта, эффективную архитектуру и практическую ориентированность.

Список литературы

1. Андреев К. С. Компьютерное зрение и машинное обучение. СПб.: Питер, 2020.
2. Букреев Д. А., Гуров А. С. Метод распознавания нарушений правил дорожного движения в режиме реального времени с помощью технологий компьютерного зрения // Университетская наука. 2025. № 1(19). С. 168-173.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2018.
4. Касаткин А. С. Нейросетевые технологии обработки изображений. СПб.: Питер, 2022.
5. Половников Е. П. Технологии машинного обучения в автоматизированных системах. М.: Академия, 2021.
6. Ткачёв Е. В. Применение нейросетевых технологий в радиолокационных системах. М.: Радиотехника, 2022.
7. Bukreiev D. O. et al. Features of the use of software and hardware of the educational process in the conditions of blended learning // AET 2020-Symposium on Advances in Educational Technology // Technology (AET 2020). SCITEPRESS. 2022. №. 2. С. 236-244.
8. Bukreiev D. Neuro-network technologies as a mean for creating individualization conditions for students learning // SHS Web of Conferences. – EDP Sciences. 2020. Т. 75. С. 04013.

**СОВРЕМЕННОЕ СОСТОЯНИЕ
РАЗВИТИЯ ТЕХНОЛОГИЙ
ПРОГНОЗИРОВАНИЯ
ФИНАНСОВЫХ ОПЕРАЦИЙ**

Соболева И. Р., Букреев Д. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: dmitriy.bukreev@mel-su.ru*

Научный руководитель: Букреев Д. А.

Введение

Прогнозирование финансовых операций занимает одно из ключевых мест в современных финансово-экономических системах, поскольку от точности и надёжности прогнозов напрямую зависят эффективность управления рисками, устойчивость финансовых организаций и обоснованность принимаемых управленческих решений. Развитие цифровых технологий, рост объёмов доступных данных и усложнение финансовых инструментов существенно трансформировали подходы к анализу и прогнозированию финансовых процессов, выдвинув на первый план интеллектуальные методы обработки информации. Финансовые операции характеризуются высокой динамичностью, стохастической природой и значительной зависимостью от внешних факторов, включая макроэкономические показатели, поведение участников рынка и глобальные экономические события. Эти особенности существенно осложняют задачу прогнозирования и ограничивают применимость традиционных статистических методов, ориентированных преимущественно на линейные зависимости и стационарные временные ряды. В условиях высокой волатильности и структурных изменений финансовых рынков такие подходы часто оказываются недостаточно устойчивыми и чувствительными к изменению входных

данных. Современное развитие технологий прогнозирования финансовых операций связано с активным внедрением методов машинного обучения и нейросетевых архитектур, способных учитывать нелинейные зависимости, взаимодействие большого числа факторов и скрытые закономерности в данных. Использование интеллектуальных моделей позволяет повысить точность прогнозов и расширить спектр решаемых задач – от краткосрочного прогнозирования транзакционной активности до анализа долгосрочных тенденций финансовых потоков. Вместе с тем применение таких методов сопровождается рядом проблем, связанных с интерпретируемостью моделей, качеством исходных данных и вычислительной сложностью алгоритмов.

Цель исследования – анализ современного состояния развития технологий прогнозирования финансовых операций и выявление основных направлений их эволюции.

Материалы и методы исследования

Материалом исследования послужили научные публикации, аналитические отчёты и результаты прикладных исследований, посвящённых прогнозированию финансовых операций. В рамках исследования учитывались как классические подходы к прогнозированию финансовых временных рядов, так и современные технологии, основанные на методах машинного обучения и нейросетевых архитектурах.

Информационную базу исследования составили теоретические и прикладные источники, отражающие эволюцию методов прогнозирования финансовых операций, включая статистические модели, регрессионные подходы, ансамблевые методы и глубокие нейронные сети. Особое внимание уделялось работам, анализирующим практическую применимость интеллектуальных моделей в условиях реальных финансовых рынков, а также ограничения, возникающие при их использовании.

Методологическая основа исследования включает совокупность аналитических методов, ориентированных на систематизацию и обобщение существующих подходов. В качестве базового применялся теоретико-аналитический метод, позволивший рассмотреть развитие технологий прогнозирования финансовых операций в динамике и выявить ключевые этапы перехода от традиционных методов к интеллектуальным моделям. Такой подход обеспечил целостное представление о современном состоянии данной области.

Для выявления особенностей и ограничений различных технологий использовался сравнительный анализ, направленный на сопоставление классических статистических методов и современных алгоритмов машинного обучения по критериям точности прогнозов, устойчивости к шумам, интерпретируемости результатов и вычислительной сложности. Это

позволило выявить сильные и слабые стороны различных подходов и определить условия их наиболее эффективного применения.

Сформированная методологическая база создаёт основу для дальнейшего анализа финансовых операций как объекта прогнозирования и рассмотрения современных технологий, применяемых для анализа и предсказания финансовых процессов, что будет отражено в последующих разделах статьи.

Результаты исследования и их обсуждение

Финансовые операции представляют собой сложный и динамичный объект прогнозирования, формируемый в условиях высокой неопределённости, многофакторного влияния и постоянных структурных изменений. В отличие от технических или производственных процессов, финансовые операции зависят не только от внутренних параметров системы, но и от широкого спектра внешних факторов, включая макроэкономическую ситуацию, регуляторные изменения, поведение участников рынка и глобальные экономические события. Это обуславливает особые требования к методам анализа и прогнозирования финансовых данных. Ключевой особенностью финансовых операций является их представление в виде временных рядов, отражающих последовательность значений финансовых показателей во времени. Такие ряды характеризуются высокой волатильностью, наличием шумов, сезонных и циклических компонентов, а также возможными резкими скачками, связанными с кризисными явлениями или изменением рыночных ожиданий. В результате финансовые временные ряды часто не удовлетворяют предпосылкам стационарности, что существенно ограничивает применение классических методов прогнозирования.

Особенности финансовых операций и временных рядов

Финансовые временные ряды обладают рядом специфических свойств, которые необходимо учитывать при построении прогнозных моделей. Одним из таких свойств является нелинейность зависимостей между показателями. Влияние факторов на финансовые операции может меняться во времени, усиливаться или ослабевать в зависимости от рыночной конъюнктуры, что делает линейные модели недостаточно гибкими для адекватного описания реальных процессов.

Другой важной особенностью является высокая степень зашумлённости данных [2]. Финансовые операции отражают совокупное воздействие большого числа случайных и трудно формализуемых факторов, включая поведенческие аспекты участников рынка. Это приводит к тому, что полезный сигнал зачастую скрыт

за значительным уровнем случайных колебаний, что усложняет его выявление и интерпретацию.

Кроме того, финансовые временные ряды подвержены структурным сдвигам, связанным с изменением экономических условий, регуляторных требований или используемых финансовых инструментов. Такие сдвиги нарушают устойчивость статистических характеристик данных и требуют от прогнозных моделей способности адаптироваться к новым условиям без существенной потери точности.

Традиционные методы прогнозирования финансовых операций, основанные на статистическом анализе и эконометрических моделях, долгое время являлись основным инструментом анализа финансовых временных рядов. К таким методам относятся регрессионные модели, модели скользящего среднего, авторегрессионные подходы и их различные модификации. Их основным преимуществом является относительная простота интерпретации и низкие вычислительные затраты. Однако применение традиционных методов связано с рядом существенных ограничений. Во-первых, многие из них предполагают линейность зависимостей и стационарность временных рядов, что редко выполняется для реальных финансовых данных. Во-вторых, такие модели слабо учитывают сложные нелинейные взаимодействия между факторами и плохо адаптируются к резким изменениям рыночной среды.

Ещё одним ограничением является ограниченная способность традиционных моделей обрабатывать большие объёмы разнородных данных [1]. Современные финансовые системы генерируют значительные массивы информации, включающие транзакционные данные, поведенческие показатели и внешние экономические индикаторы. Использование таких данных в рамках классических подходов требует значительного упрощения модели, что может приводить к потере важной информации и снижению качества прогнозов.

Таким образом, анализ финансовых операций как объекта прогнозирования показывает, что их сложность и динамичность существенно ограничивают применимость традиционных методов. Это создаёт предпосылки для активного внедрения современных технологий прогнозирования, способных учитывать нелинейность, адаптивность и многофакторность финансовых процессов, что будет рассмотрено в следующем разделе статьи.

Современный этап развития технологий прогнозирования финансовых операций характеризуется смещением акцента от строго формализованных статистических моделей к интеллектуальным методам анализа данных. Это обусловлено как ростом вычислительных возможностей, так и увеличением объёмов и разнообразия финансовой информации, доступной для анализа [4]. В таких условиях прогноиро-

вание перестаёт быть задачей аппроксимации временных рядов и всё чаще рассматривается как задача выявления скрытых закономерностей в сложных многофакторных системах. Ключевой особенностью современных технологий является их способность работать с нелинейными зависимостями, учитывать влияние большого числа признаков и адаптироваться к изменяющимся условиям финансовой среды. Это позволяет повысить точность прогнозов и расширить область их применения, однако одновременно порождает новые методологические и практические проблемы.

Методы машинного обучения заняли прочное место в задачах прогнозирования финансовых операций благодаря своей гибкости и способности обрабатывать большие массивы данных. В отличие от традиционных моделей, они не требуют явного задания формы зависимости между входными и выходными параметрами, что позволяет учитывать сложные и нелинейные взаимосвязи в финансовых данных. Широкое распространение получили деревья решений, методы опорных векторов и ансамблевые подходы, такие как случайные леса и градиентный бустинг. Ансамблевые методы позволяют повысить устойчивость прогнозов за счёт комбинирования нескольких моделей и снижения влияния отдельных ошибок. Это особенно важно в условиях высокой зашумлённости финансовых данных и нестабильности рыночных процессов. В то же время применение методов машинного обучения в финансовом прогнозировании требует тщательной подготовки данных и выбора признаков. Качество прогнозов в значительной степени зависит от репрезентативности обучающей выборки и корректности учёта временной структуры данных. Нарушение этих условий может приводить к переобучению моделей и снижению их прогностической ценности при использовании на новых данных [3].

Нейросетевые модели являются одним из наиболее активно развивающихся направлений в области прогнозирования финансовых операций. Их ключевым преимуществом является способность автоматически выявлять сложные зависимости в данных без явного задания правил и закономерностей. Особенно перспективными оказались рекуррентные нейронные сети и их модификации, ориентированные на работу с последовательностями и временными рядами. Применение нейросетевых архитектур позволяет учитывать долгосрочные зависимости и взаимодействие множества факторов, что особенно важно для анализа финансовых операций, подверженных влиянию накопленных эффектов и инерционных процессов. Глубокие нейронные сети находят применение как в задачах краткосрочного прогнозирования транзакционной активности, так и при анализе долгосрочных тенденций финансовых потоков.

Однако использование нейросетевых моделей связано с рядом ограничений. Высокая вычислительная сложность, потребность в значительных объёмах данных и ограниченная интерпретируемость результатов затрудняют их практическое внедрение в ряде финансовых систем. Кроме того, нейросетевые модели чувствительны к изменениям структуры данных, что требует регулярного обновления и повторного обучения моделей. Несмотря на значительные достижения в области интеллектуальных технологий прогнозирования, их практическое применение в финансовых операциях сопровождается рядом проблем. Одной из ключевых является проблема интерпретируемости моделей [5]. В финансовой сфере важна не только точность прогноза, но и возможность объяснить полученные результаты, что затруднено при использовании сложных нейросетевых архитектур.

Другим существенным ограничением является риск переобучения моделей, особенно при работе с ограниченными или нестабильными данными. Финансовые рынки подвержены структурным изменениям, что снижает устойчивость моделей, обученных на исторических данных. В таких условиях требуется регулярная адаптация прогнозных алгоритмов и контроль их актуальности. Дополнительную сложность представляет высокая вычислительная нагрузка интеллектуальных моделей и требования к инфраструктуре. Это ограничивает их использование в системах, где необходима высокая скорость принятия решений или имеются ограничения по вычислительным ресурсам.

Таким образом, современные технологии прогнозирования финансовых операций обладают значительным потенциалом, однако их эффективное применение требует взвешенного подхода, учитывающего специфику финансовых данных, требования к интерпретируемости и устойчивости моделей. Понимание этих ограничений является важной предпосылкой для формирования обоснованных выводов о текущем состоянии и перспективах развития прогнозных технологий.

Заключение

Проведённый анализ показал, что прогнозирование финансовых операций на современном этапе развития финансово-экономических систем представляет собой сложную и многоаспектную задачу, требующую применения гибких и адаптивных аналитических инструментов. Финансовые операции как объект прогнозирования характеризуются высокой волатильностью, нелинейностью зависимостей, зашумлённостью данных и существенной зависимостью от внешних факторов, что ограничивает эффективность традиционных статистических и эконометрических подходов. Рассмотрение эволюции методов прогнозирования позволило установить,

что классические модели, несмотря на их интерпретируемость и относительную простоту, в значительной степени уступают современным технологиям в условиях роста объемов данных и усложнения финансовых процессов. Ограничения, связанные с предпосылками стационарности и линейности, делают традиционные методы недостаточно устойчивыми при анализе реальных финансовых временных рядов. Современные технологии прогнозирования финансовых операций, основанные на методах машинного обучения и нейросетевых архитектурах, демонстрируют высокий потенциал за счёт способности учитывать нелинейные зависимости и взаимодействие большого числа факторов. Ансамблевые методы и глубокие нейронные сети позволяют повысить точность прогнозов и расширить спектр решаемых задач, однако их применение сопровождается новыми проблемами, включая интерпретируемость результатов, риск переобучения и высокие требования к качеству данных и вычислительным ресурсам.

Таким образом, современное состояние развития технологий прогнозирования финансовых операций характеризуется переходом от универсальных моделей к специализированным интеллектуальным решениям, ориентированным на конкретные задачи и условия применения. Эффективное использование таких технологий требует взвешенного подхода, сочетающего точность прогнозирования, устойчивость моделей и возможность их практической интерпретации. Полученные выводы подтверждают, что дальнейшее развитие прогнозных технологий будет связано с интеграцией интеллектуальных методов, совершенствованием подходов к работе с данными и развитием гибридных моделей, учитывающих как статистические, так и интеллектуальные компоненты анализа.

Список литературы

1. Букреев Д. А., Чайка А. А. Перспективы использования технологии искусственных нейронных сетей в образовательных учреждениях // Современные проблемы геометрического моделирования и информационные технологии: материалы II Межрегиональной научно-практической конференции преподавателей и студентов, посвященной 60-летию образования Мелитопольской школы прикладной геометрии, Мелитополь, 28 мая 2024 года. Мелитополь: Мелитопольский государственный университет, 2024. С. 105-110. EDN: QRDJDP.
2. Найдыш А. В., Букреев Д. А. Интеллектуальные технологии как средство повышения качества подготовки ИТ-специалистов эксплуатации ветровых электростанций // Экосистемы. 2024. № 40. С. 14-25. DOI: 10.29039/2413-1733-2024-40-14-25.
3. Bahadir E. Using Neural Network and Logistic Regression Analysis to Predict Prospective Mathematics Teachers' Academic Success upon Entering Graduate Education // Educational Sciences: Theory and Practice. 2016. Vol. 16. No. 3. P. 943-964.
4. Bukreiev D. Neuro-network technologies as a mean for creating individualization conditions for students learning // SHS Web of Conferences. EDP Sciences. 2020. Vol. 75. P. 04013.
5. Cox A. M. Exploring the impact of Artificial Intelligence and robots on higher education through literature-based design fictions // International Journal of Educational Technology in Higher Education. 2021. Vol.18. No. 1. S. 3.

ЭВОЛЮЦИЯ АДАПТИВНОГО ДИЗАЙНА

Стрельченко О. В., Олейник Н. П.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: v.strelchenko04@mail.ru,
nata.oleynik.2014@mail.ru

Постановка проблемы. Всестороннее изучение процесса развития адаптивного дизайна, начиная с его зарождения и до современного состояния направлено на систематизацию знаний и выявление тенденций, которые смогут помочь разработчикам и дизайнерам эффективнее применять адаптивные решения с учётом современных требований и технологических возможностей.

Материалы и методы исследования

За последние годы адаптивный дизайн претерпел значительные изменения, что отражается в многочисленных научных и практических исследованиях. Современные работы всё чаще сосредотачиваются не только на технических аспектах адаптивности, но и на пользовательском опыте, кроссплатформенной совместимости и эффективности взаимодействия. Кристофер Коишагава акцентирует внимание на том, что «в наши дни адаптивный веб-дизайн воспринимается как нечто само собой разумеющееся. Когда мы заходим на сайт, мы ожидаем, что он будет корректно отображаться на всех наших устройствах, независимо от размера экрана. Но потребовалось много времени, чтобы прийти к этому, и разработчики перепробовали множество способов адаптировать сайты под разные размеры экранов, прежде чем остановились на адаптивном веб-дизайне» [1].

Авторы в своей статье утверждают, что «Создание высокопроизводительных веб-сайтов, особенно адаптивных, требует комплексного подхода к оптимизации. Медленная загрузка страницы негативно влияет на пользовательский опыт и SEO-показатели. Для достижения оптимальной производительности необходимо учитывать, как клиентскую, так и серверную стороны» [9].

Так в своей статье Лемешко Клавдия подчеркивает, что «адаптивная верстка представляет собой обширный набор инструментов, которые позволяют осуществить настройку конфигурации сайта с учетом потребностей различных устройств. Это универсальный и наиболее эффективный метод веб-дизайна» [2]. А Йони Корпи в своей работе писал, что «адаптивный веб-дизайн – это новая парадигма в разработке веб-сайтов. Он связан с тем, что веб-сайты больше не отображаются только на мониторах определенного размера, но и на экранах бесчисленного множества различных размеров, начиная от крошечных мобильных телефонов и заканчи-

вая 50-дюймовыми широкоформатными телевизорами. В отличие от традиционных веб-сайтов фиксированной ширины, хорошие адаптивные веб-сайты перестраиваются под экраны практически любого размера, при этом содержание сайта остается разборчивым, а его контекст – соответствующим» [3]. Одним из ключевых трендов последних исследований является интеграция машинного обучения и искусственного интеллекта для автоматической подстройки интерфейсов под индивидуальные потребности пользователей и контекст использования. Исследования показывают, что эффективный адаптивный дизайн сегодня – это комплексная система, объединяющая гибкую верстку, интеллектуальные алгоритмы и продуманную архитектуру интерфейсов, способную учитывать разнообразие устройств.

Целью исследования является анализ исторического контекста появления адаптивного дизайна, рассматривая предпосылки и вызовы, которые обусловили необходимость создания гибких интерфейсов. Рассмотрим роль развития мобильных устройств и различных экранных форматов в формировании требований к дизайну и проследим, как адаптивный дизайн трансформировался из частного решения в универсальный стандарт веб-разработки.

Результаты исследования и их обсуждение

Современный мир цифровых технологий характеризуется постоянным и стремительным развитием устройств, через которые пользователи получают доступ к информации и онлайн-сервисам. Смартфоны, планшеты, ноутбуки, а также умные телевизоры и смарт-часы создают огромное разнообразие экранов с разными размерами, разрешениями и особенностями отображения. Специфика каждого устройства и разнообразие пользовательских сценариев вынуждают разработчиков искать баланс между

универсальностью и оптимизацией под конкретные платформы. Анализ процессов, этапов развития и ключевых тенденций в области адаптивного дизайна становится актуальной задачей для специалистов, стремящихся создавать современные и эффективные решения.

Веб-дизайн зародился вместе с появлением первого в истории веб-сайта (<https://info.cern.ch/>, рисунок), созданного Тимом Бернерсом-Ли в 1991 году. Этот сайт представлял собой простую текстовую страницу без графики, ориентированную исключительно на передачу информации [1].

В то время конструкция страниц была ограничена возможностями HTML, который позволял создавать лишь статические элементы с минимальной структуризацией и форматированием. Дизайн как таковой почти отсутствовал, поскольку основная задача заключалась в предоставлении доступа к контенту [1].

Первые веб-страницы выглядели одинаково на всех устройствах, так как не учитывалось разнообразие экранов и разрешений. Страницы были рассчитаны на мониторы с фиксированным размером (800x600 или 1024x768 пикселей), а единственным способом визуального оформления были базовые теги для выравнивания текста и простого форматирования. Цвета и изображения использовались редко, что объяснялось ограниченной пропускной способностью сетей и отсутствием мощных инструментов для создания графического контента.

С ростом популярности интернета и появлением первых браузеров, таких как Mosaic и Netscape Navigator, веб-дизайнеры начали экспериментировать с таблицами, фреймами и стилями. Появление CSS в середине 1990-х открыло новые возможности для визуального оформления сайтов, позволяя отделить контент от презентации. Однако дизайн по-прежнему оставался статичным и не адаптировался под различные устройства.

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Вид первого веб-сайта

На начальном этапе развития веб-дизайна сайты создавались преимущественно с фиксированной шириной, что означало строгие рамки для размещения контента. Это позволяло дизайнерам контролировать внешний вид страниц, гарантируя, что элементы интерфейса всегда будут располагаться именно там, где задумано.

Главным преимуществом фиксированных сайтов была простота реализации и предсказуемость отображения. Размещение текстов, изображений и навигационных элементов оставалось неизменным вне зависимости от устройства, что облегчало разработку и поддержку. Однако с появлением множества разнообразных устройств с различными размерами экранов – от небольших мобильных телефонов до широкоформатных мониторов – фиксированные сайты стали сталкиваться с серьезными ограничениями.

2007 год стал поворотным моментом в истории веб-дизайна – Apple представила iPhone, изменивший представления о мобильном веб-серфинге. Впервые пользователи получили полноценный браузер на мобильном устройстве, способный отображать «настоящие» веб-страницы, а не их упрощенные WAP-версии [4].

Тогда же компания Apple разработала метатеги viewport, чтобы указывать браузерам на то, в каком масштабе необходимо отображать пользователю видимую область веб-страницы [5].

Неадаптивные сайты теряли актуальность с ростом числа пользователей мобильных устройств и разнообразием экранов, что и послужило толчком к развитию принципов адаптивного дизайна, позволяющего гибко подстраиваться под любые размеры экрана и улучшать взаимодействие с сайтом.

Первоначально дизайнеры и разработчики реагировали созданием отдельных мобильных версий сайтов (часто с доменами m.website.com). Такие сайты усложняли поддержку и обновление [1].

В 2010 году дизайнер Итан Маркотт представил концепцию «отзывчивого веб-дизайна» (responsive web design) в своей статье на A List Apart. Его подход предлагал создавать один сайт, который адаптировался бы к различным размерам экрана с помощью гибких сеток, гибких изображений и медиа-запросов CSS [5].

Гибкие сетки (fluid grids) вытеснили традиционные фиксированные макеты, основанные на точных пиксельных значениях. Вместо фиксированных ширин колонок и блоков контента дизайнеры стали использовать относительные единицы измерения – проценты, единицы em и rem. Это позволило элементам страницы плавно масштабироваться в зависимости от ширины экрана, создавая гибкую структуру, которая адаптируется к изменениям размеров окна браузера или устройства пользователя [6].

Параллельно с гибкими сетками развивались гибкие медиаресурсы, в первую очередь изображения и видео. В классическом подходе размеры медиафайлов были фиксированы, что приводило к плохой производительности и некорректному отображению на устройствах с маленькими экранами. Гибкие медиа обеспечили возможность изображениям занимать только необходимую часть экрана, используя свойства CSS, такие как max-width: 100%, что означает, что изображение не выходит за пределы контейнера и масштабируется вместе с остальным содержимым [6].

Медиа-запросы (media queries) же стали революционной возможностью для CSS, позволяя применять различные стили в зависимости от характеристик устройства или окна браузера – ширины, высоты, разрешения экрана и других параметров. Это дало дизайнерам мощный инструмент для создания адаптивных интерфейсов: можно было задавать несколько вариантов оформления под разные условия отображения, переключая структуры, шрифты, размеры и расположение элементов [6].

Развитие адаптивного дизайна началось с популяризации Bootstrap, который предложил разработчикам готовую систему 12-колоночных сеток и сделал мобильную адаптацию массовым стандартом. На смену тяжеловесным фреймворкам пришел Flexbox, совершивший революцию в выравнивании элементов и гибком распределении пространства внутри одномерных структур, таких как меню или карточки [7].

Следующим важным этапом стало внедрение CSS Grid – полноценной двумерной системы макетов, которая позволила создавать сложные интерфейсы с сетками любой сложности на чистом CSS. Финальной точкой в этой эволюции стали Container Queries, которые перенесли логику адаптивности с размеров всего экрана на размеры конкретного родительского блока, позволив компонентам самостоятельно подстраиваться под любое место в интерфейсе и сделав дизайн по-настоящему модульным [7].

Появление и стремительное развитие искусственного интеллекта стало одним из ключевых факторов, повлиявших на трансформацию подходов к веб-дизайну, в частности – на эволюцию адаптивного дизайна. Внедрение интеллектуальных технологий открыло новые возможности для автоматизации, персонализации и динамической оптимизации интерфейсов, что позволило сделать процессы адаптации сайтов под различные устройства более точными, эффективными и ориентированными на потребности пользователя [8].

Алгоритмы машинного обучения анализируют данные о посетителях, особенностях их устройств и поведении на сайте, что делает возможной динамическую адаптацию структуры и визуальных элементов интерфейса в режиме реального времени. Такой подход способствует

созданию более интуитивного и персонализированного пользовательского опыта. Сайт корректно отображается на любом устройстве, повышается его удобство и визуальная согласованность, что напрямую влияет на эффективность взаимодействия и увеличивает вероятность совершения целевого действия. Таким образом, применение искусственного интеллекта в адаптивном веб дизайне представляет собой шаг к созданию интеллектуальных, гибких цифровых решений нового поколения [8].

Заключение

Проведённое исследование позволяет сделать вывод о том, что адаптивный дизайн прошёл значительный путь эволюции, от решения узких технических задач по корректному отображению контента на различных устройствах до становления полноценной концепции, определяющей стандарты современной веб разработки. Его развитие закономерно связано с ростом разнообразия мобильных устройств, экранных форматов и способов взаимодействия пользователей с цифровыми платформами. На современном этапе адаптивный дизайн характеризуется переходом от фиксированных шаблонов к интеллектуальным системам, способным динамически подстраиваться под индивидуальные потребности пользователя и контекст использования. В стратегической перспективе адаптивный дизайн рассматривается как универсальный принцип проектирования цифровых продуктов, обеспечивающий их гибкость, устойчивость и соответствие актуальным технологическим тенденциям.

Список литературы

1. Koishigawa K. A Brief History of Responsive Web Design. URL: <https://www.freecodecamp.org/news/a-brief-history-of-responsive-web-design/> (дата обращения: 15.12.2025).
2. Лемешко К. Г., Калайда А. В. Современные тенденции веб-дизайна и адаптивной вёрстки // Научный аспект. 2019. № 2. С. 1634-1639. [Электронный ресурс]. URL: https://na-journal.ru/pdf/nauchnyi_aspekt_2-2019_t13_web.pdf#page=74 (дата обращения: 15.12.2025).
3. Joni Korpi. Adaptive web design. As applied to the design process of a web application. [Электронный ресурс]. URL: https://www.theseus.fi/bitstream/handle/10024/40696/Korpi_Joni.pdf (дата обращения: 11.12.2025).
4. Эволюция веб-дизайна: от HTML-страниц к адаптивным системам. [Электронный ресурс]. URL: <https://sky.pro/wiki/profession/istoriya-i-razvitie-veb-dizajna/> (дата обращения: 15.12.2025).
5. meta viewport. Как он работает? [Электронный ресурс]. URL: <https://itchief.ru/html-and-css/viewport-meta-tag> (дата обращения: 18.12.2025).
6. Егорова И. Н., Толстая Е. А. Исследование принципов адаптивного веб-дизайна. URL: <https://cyberleninka.ru/article/n/issledovanie-printsipov-adaptivnogo-veb-dizajna/viewer> (дата обращения: 15.12.2025).
7. Руководство по реализации отзывчивого дизайна в 2023 году. [Электронный ресурс]. URL: <https://habr.com/ru/companies/ruvds/articles/718700/> (дата обращения: 17.12.2025).
8. Использование ии для создания адаптивного веб-дизайна. [Электронный ресурс]. URL: <https://smm-agentstvo.ru/blog/ispolzovanie-ii-dlia-sozdaniia-adaptivnogo-veb-dizajna/> (дата обращения: 10.12.2025).

9. Стрельченко О. В., Олейник Н. П. Адаптивный дизайн и производительность / Материалы XVII Международного студенческого научного форума, 2025. [Электронный ресурс]. URL: <https://scienceforum.ru/2025/article/2018037504> (дата обращения: 15.12.2025).

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ БАЛАНСА ИГРОВЫХ ХАРАКТЕРИСТИК В АВТОМАТИЗИРОВАННЫХ АНАЛОГАХ НАСТОЛЬНЫХ ИГР

Сушинский И. В., Лебедев В. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: v.lebedev@melsu.ru

Научный руководитель: Лебедев В. А.

Введение

Игровая индустрия последних лет демонстрирует рост интереса к мобильным платформам и мессенджерам. Текстовые квесты и адаптации настольных игр на базе чат-ботов (в том числе в Telegram) привлекают пользователей доступностью: нет необходимости устанавливать дополнительное ПО, игра интегрирована в привычную среду общения [1]. Особенностью данного сегмента является низкий порог входа для разработчиков и мгновенное распространение через механизмы вирусного маркетинга в групповых чатах.

Перенос классических настольных игр (например, «Бункер» – игра на выживание и социальную дедукцию) в цифровой формат создаёт технические сложности. В физической версии баланс и уникальность игровых ситуаций достигаются конечным набором карт в колоде. Взятая карта покидает набор, повторное её появление невозможно.

Механика физической колоды обеспечивает несколько критически важных свойств игрового процесса. Конечность ресурсов создаёт информационное преимущество для внимательных игроков: запоминание выбывших карт позволяет вычислять вероятности появления оставшихся элементов. Детерминированность распределения исключает статистические аномалии – невозможна ситуация, когда в десяти последовательных раздачах доминирует одна категория карт.

В программной реализации со стандартными генераторами псевдослучайных чисел (ГПСЧ) возникает риск нарушения этого принципа [2]. Без алгоритмического контроля возможны дублирование уникальных ролей (два персонажа с одинаковой биологией или профессией) или искажения вероятностного распределения характеристик. Такие ошибки разрушают игровую логику и снижают вовлечённость пользователей.

Стандартные генераторы псевдослучайных чисел, основанные на алгоритмах линейного

конгруэнтного метода или вихря Мерсенна, обеспечивают статистическую случайность последовательности, но не гарантируют уникальность элементов в ограниченной выборке. При наивной реализации вероятность коллизии растёт экспоненциально с увеличением числа участников. Для колоды из N карт вероятность дублирования при k выборках описывается парадоксом дней рождения и достигает 50 процентов уже при k примерно равном корню квадратный из N [7].

Цель исследования – разработать и описать алгоритм процедурной генерации игрового контента, который обеспечивает математически строгий баланс характеристик и воспроизводит механику «выборки без возвращения» в Telegram-боте.

Поставленная цель предполагает решение комплекса взаимосвязанных задач. Необходимо обеспечить изоморфизм вероятностных свойств между физической и цифровой реализацией, то есть для любого подмножества карт вероятность его появления в цифровой версии должна совпадать с вероятностью выбора соответствующих физических карт. Важно спроектировать архитектуру хранения состояния, устойчивую к сбоям и обеспечивающую консистентность данных при параллельных запросах от игроков одной сессии.

Материалы и методы исследования

Программная архитектура чат-бота основана на методах алгоритмизации стохастических процессов и технологиях работы с структурированными данными в Python. Методологическая база: встроенные механизмы сериализации объектов для управления состоянием сессии, применение регулярных выражений для синтаксического анализа и преобразования строк. Реализована логика конечных автоматов для корректного переключения фаз игры и целостности данных между запросами.

В качестве базовой методологии выбран принцип чистых функций с иммутабельными структурами данных. Каждая операция генерации создаёт новое состояние игровой сессии, не модифицируя предыдущее, что упрощает отладку и позволяет реализовать механизм отката транзакций при ошибках. Для управления конкурентным доступом применяется блокировка на уровне файловой системы, предотвращающая состояние гонки при одновременных запросах игроков.

Материал исследования – корпусы текстовых данных в JSON-структурах с массивами категориальных переменных: профессий, биологических характеристик, карт здоровья и предметов. Данные имеют жёсткие перекрёстные зависимости, требующие валидации при генерации. Для создания игровых профилей используется выборка без возвращения через ди-

намическое исключение элементов из списков. Этот подход с условной логикой корректировки параметров (синхронизация возраста и стажа) моделирует замкнутую экономику ресурсов, исключая коллизии и противоречия в контенте.

Структура данных организована по принципу нормализованных реляционных таблиц, адаптированных к формату JSON. Каждая категория характеристик представлена массивом объектов с атрибутами. Карты здоровья имеют иерархическую структуру: корневой объект описывает заболевание, вложенный массив содержит возможные стадии с различной степенью тяжести. Данные прошли экспертную валидацию на предмет логической непротиворечивости.

Результаты исследования и их обсуждение

Архитектура программного комплекса – событийно-ориентированная система, интегрированная в Telegram. Основная логика генерации контента вынесена в отдельный модуль на Python. В отличие от стандартных чат-ботов без сохранения состояния, данная реализация фиксирует контекст каждой игровой сессии.

Событийная модель построена на паттерне наблюдатель-подписчик. Каждое действие пользователя генерирует событие, которое обрабатывается соответствующим обработчиком без блокирования основного цикла приложения. Разделение логики генерации контента в отдельный модуль следует принципу единственной ответственности: модуль взаимодействия с API мессенджера отвечает только за маршрутизацию сообщений, модуль генерации – исключительно за создание игрового контента.

Модель данных: каждая активная партия представлена уникальным JSON-объектом. Этот файл – динамическая база данных, хранящая состояние «колоды» игровых карт. Принципы построения: генерация общей среды и персонафикация игроков.

JSON-объект сессии включает метаданные игры: уникальный идентификатор партии, временную метку создания, список участников с их Telegram ID, текущую фазу игрового процесса. Динамическая часть содержит актуальное состояние колоды – массивы доступных карт для каждой категории. Важным аспектом является атомарность операций: каждое изменение состояния сопровождается немедленной записью на диск, что гарантирует консистентность даже при аварийном завершении процесса.

1. Алгоритмическая реализация генерации среды

Функция `create_bunker` инициализирует игровой мир и создаёт контекст выживания. Алгоритм загружает список карт из файла-шаблона, выбирает случайные элементы и удаляет их

из доступных опций (ключевой момент для баланса) [4]. Программная реализация этапа:

Детализация алгоритмической логики включает несколько критических этапов. Загрузка файла сессии выполняется с обработкой исключений: если файл повреждён или содержит невалидный JSON, система создаёт новый объект сессии с предупреждением администратору. Выборка катастрофы реализована через метод `random.choice`, использующий вихрь Мерсенна в качестве базового ГПСЧ [5]. После выбора элемент немедленно удаляется из массива методом `remove`.

Загрузка состояния: система читает файл `game_id.json`.

Выборка катастрофы: метод `random.choice()` определяет тип угрозы, выбранный элемент удаляется методом `remove()`. Это исключает коллизии условий в партии.

Формирование ресурсов: в цикле выбираются предметы для бункера. Исключение выбранных элементов на каждой итерации моделирует физическое убывание, повышая ценность оставшихся предметов.

Такой подход обеспечивает уменьшение множества `Sstore` с каждой итерацией, предотвращая дубликаты предметов.

$$|S_{i+1}| = |S_i| - 1$$

Математическая формализация процесса: пусть исходное множество `S` содержит `N` элементов. После `k` выборов мощность множества становится равной `N` минус `k`. Вероятность выбора конкретного элемента на `i`-й итерации равна единица делённая на `N` минус `i` плюс единица, что точно соответствует вероятности извлечения карты из физической колоды с уменьшающимся количеством карт.

Важно отметить, что применение метода выборки без возвращения обеспечивает не только уникальность элементов, но и сохраняет статистические свойства распределения, соответствующие физической реализации игры. Это критически важно для поддержания справедливости игрового процесса и доверия пользователей к системе.

2. Комплексная генерация ролевых характеристик

Наиболее сложный этап – распределение характеристик персонажа с учётом логических зависимостей между параметрами. Реализованная логика:

Сложность данного этапа обусловлена необходимостью согласования множественных ограничений целостности. Каждый атрибут персонажа существует в отдельном семантическом пространстве, но между пространствами установлены жёсткие связи. Профессия накладывает ограничения на минимальный возраст,

биологические характеристики влияют на допустимые виды хобби, состояние здоровья коррелирует с возрастными группами.

Атомарная выборка: профессия, биологические характеристики, хобби и факты выбираются без возвращения. Это гарантирует уникальность игроков в партии.

Процедурная валидация: для реализма стаж зависит от возраста. Алгоритм собирает строку биологии, извлекает возраст регулярными выражениями, генерирует стаж в диапазоне `k0`, Возраст – 18]. При нулевом стаже генерируется значение в месяцах с помощью функции склонения.

Алгоритм извлечения возраста использует регулярное выражение, сканирующее строку биологии на наличие паттерна «число лет». После извлечения числового значения выполняется валидация диапазона: возраст должен находиться в интервале от 18 до 80 лет согласно правилам игры. Специальная обработка нулевого стажа предусматривает конвертацию в месяцы: генерируется значение от 1 до 11, затем применяется функция склонения существительного «месяц» в соответствии с правилами русского языка.

Иерархическая генерация здоровья: метод `pop()` извлекает объект болезни целиком, затем выбирается конкретная стадия. Это предотвращает ситуацию, когда два игрока получают одну болезнь с разными стадиями (противоречие механике уникальности).

Метод `pop` обеспечивает атомарность операции выборки: элемент извлекается из массива и одновременно удаляется, что исключает необходимость отдельного вызова `remove` и предотвращает условие гонки при параллельных запросах. После извлечения объекта болезни система обращается к вложенному массиву стадий. Выбор стадии выполняется независимым вызовом `random.choice`, поскольку стадии одного заболевания не являются уникальными между партиями.

Процесс генерации характеристик построен таким образом, чтобы обеспечить максимальное разнообразие игровых ситуаций при сохранении логической целостности. Каждый персонаж представляет собой уникальную комбинацию параметров, что способствует повышению вовлечённости и стратегической глубины игрового процесса.

Комбинаторная сложность пространства возможных персонажей достигает астрономических значений. При типичной конфигурации игры с 50 профессиями, 40 биологическими характеристиками, 30 хобби и 25 картами здоровья количество теоретически возможных уникальных комбинаций превышает 10 в степени 9. Это обеспечивает практически бесконечную реиграбельность: вероятность получить идентичную комбинацию в двух независимых партиях пренебрежимо мала.

3. Оценка эффективности и игрового баланса

Тестирование проведено на выборке из 1000 симуляций игровых сессий. Критерии оценки: отсутствие дубликатов уникальных ролей и семантическая корректность профилей.

Методология тестирования включала автоматизированную генерацию игровых сессий с различным количеством участников: от минимального состава в 4 игрока до максимального в 24 человека. Каждая симуляция выполняла полный цикл создания бункера и распределения ролей, затем результаты проходили серию проверок на уникальность множеств, валидацию перекрёстных зависимостей и анализ семантической корректности.

Анализ результатов: метод выборки без возвращения обеспечивает «входную случайность». Вероятность P получения профессии x для k -го игрока при общем количестве профессий N динамически изменяется:

$$P(x_k) = \frac{1}{N - (k - 1)}$$

где $(k - 1)$ – количество выданных ролей. С каждым игроком пространство вариантов сужается, но вероятность получения любой из оставшихся ролей равномерна, как при физической раздаче карт.

Данное свойство алгоритма является прямым следствием теоремы о гипергеометрическом распределении. В отличие от схемы выборки с возвращением, где вероятности остаются константными, предлагаемый метод моделирует конечную популяцию без пополнения. Эмпирическая верификация теоретических расчётов выполнена методом Монте-Карло: для каждой профессии вычислялась частота появления в 1000 симуляциях, затем применялся критерий хи-квадрат для проверки гипотезы о равномерности распределения.

Система устойчива к логическим ошибкам: в 100% случаев числовые параметры (возраст, стаж, длительность хобби) находились в допустимых диапазонах, что подтверждает эффективность процедурных ограничений.

Дополнительно было проведено сравнительное исследование производительности различных алгоритмов генерации, которое подтвердило оптимальность выбранного подхода. Время обработки одного запроса составляет менее 50 мс, что обеспечивает мгновенный отклик системы с точки зрения пользователя.

Сравнительный анализ включал три альтернативных подхода к генерации [3]. Первый – наивная реализация с проверкой дубликатов через множество: каждая выборка добавляла элемент в набор, при коллизии выборка повторялась. Второй подход – предварительное перемешивание всей колоды алгоритмом

Фишера-Йейтса с последовательной раздачей. Предлагаемый метод с выборкой без возвращения показал оптимальное соотношение времени выполнения и потребления памяти.

4. Практические аспекты внедрения и оптимизации

Интеграция с мессенджером Telegram осуществляется через официальный Bot API с использованием механизма вебхуков по протоколу HTTPS [6]. В отличие от метода длительного опроса, данная архитектура инициирует передачу данных только при наступлении события, что существенно снижает холостую нагрузку на сервер и сетевой трафик. Обработка входящих запросов происходит асинхронно, что позволяет обслуживать множество параллельных игровых сессий без блокировки основного потока выполнения.

Архитектура вебхуков требует настройки обратного прокси-сервера, обеспечивающего терминацию SSL и переадресацию запросов к приложению Python. Каждый входящий запрос от API Telegram содержит JSON-объект с информацией о событии: тип события, идентификатор пользователя, текст сообщения, метаданные чата. Приложение парсит этот объект, определяет текущую фазу игровой сессии для данного чата, вызывает соответствующий обработчик.

Нагрузочное тестирование в условиях эксплуатации проводилось на виртуальном сервере с конфигурацией: 4 ядра CPU и 16 ГБ RAM. Результаты показали высокую эффективность оптимизации кода: загрузка процессорных мощностей не превышала 30% при одновременной поддержке 50 активных игровых сессий. Среднее время отклика системы оставалось в пределах комфортных значений UX, что подтверждает масштабируемость решения и корректность выбранного архитектурного подхода.

Сценарий нагрузочного тестирования моделировал реалистичное поведение пользователей: создание новых игровых комнат с различными интервалами, одновременные запросы генерации персонажей от всех участников партии.

Заключение

Разработано и программно реализовано алгоритмическое обеспечение для автоматизированного аналога настольной игры «Бункер». Метод выборки без возвращения с хранением состояния колоды в JSON эффективен для решения проблемы дублирования игровых сущностей.

Функции `create_bunker` и `create_person` решают задачу цифровой имитации физических компонентов игры: обеспечивают уникальность ситуаций и реиграбельность, логическую непротиворечивость характеристик через перекрёстную валидацию параметров, математиче-

ски обоснованный баланс для соревновательной механики социальной дедукции.

Достигнутые результаты имеют методологическое значение для проектирования аналогичных систем. Продемонстрирована применимость классических алгоритмических подходов теории вероятностей к современным задачам разработки игрового программного обеспечения. Архитектурное решение с хранением состояния в JSON-файлах демонстрирует оптимальный баланс между простотой реализации, производительностью и надёжностью для класса приложений с умеренной нагрузкой.

Результаты показывают целесообразность применения таких алгоритмических подходов при разработке чат-ботов для настольных игр: качественные цифровые адаптации с минимальными требованиями к клиентскому ПО.

Список литературы

1. Гордеев Д. А., Ковалева Т. М. Разработка чат-ботов для мессенджера Telegram на языке Python // E-Scio. 2021. № 4 (55). С. 526-533.
2. Пименов М. А. Практики балансирования компьютерных игр // International Journal of Open Information Technologies. 2020. Т. 8, № 11. С. 53-60.
3. Сысоева А. А. Анализ алгоритмов процедурной генерации уровней в видеоиграх // StudNet. 2021. Т. 4, № 6. С. 1245-1252.
4. Епифанов А. С. Гибридный алгоритм процедурной генерации подземелий для 2D-игр // Инновации и инвестиции. 2021. № 5. С. 235-238.
5. Кекеев К. В. Анализ генераторов псевдослучайных чисел // Вестник современных исследований. 2019. № 1.14 (28). С. 22-25.
6. Рубцов А. А. Разработка чат-бота для мессенджера Telegram: практические аспекты // Вестник науки. 2021. Т. 4, № 9 (42). С. 53-57.
7. Гмурман В. Е. Теория вероятностей и математическая статистика: учебное пособие для вузов. 12-е изд. М.: Юрайт, 2024. 479 с.

ЦИФРОВЫЕ ТЕХНОЛОГИИ В СФЕРЕ ОБРАЗОВАНИЯ ЗАПОРОЖСКОЙ ОБЛАСТИ

Тарасенко И. В.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ilyatarasenko2020@gmail.com*

Научный руководитель: Ступницкий В. С.

Введение

Современный этап развития общества характеризуется стремительной цифровизацией всех сфер жизнедеятельности, и в соответствии с Распоряжением Правительства РФ от 2 декабря 2021 г. № 3427-р «Об утверждении стратегического направления в области цифровой трансформации образования, относящейся к сфере деятельности Министерства просвещения РФ» [1], образование не является исключением. Вне-

дрение цифровых технологий в образовательную среду открывает множество возможностей улучшения не только имеющихся педагогических методик, но и развитие новых, а также повышает качество предоставляемых образовательных услуг.

В рамках исследования в данной статье будут рассмотрены ключевые направления цифровизации, выявлены основные проблемы и препятствия, а также предложены конкретные рекомендации по оптимизации этого процесса с учетом специфики региона. Особое внимание будет уделено сравнению опыта цифровизации в Запорожской области с другими новыми регионами, что позволит выявить наиболее эффективные стратегии и практики, направленной на повышение конкурентоспособности региона и обеспечение равных возможностей для всех обучающихся.

Цель исследования – на основе анализа имеющихся данных по цифровизации образовательной сферы Запорожской области разработать новую стратегию внедрения, более простую для пользователей.

Материалы и методы исследования

В основе предлагаемой стратегии лежит собранная статистика в период с марта 2025-го года по декабрь 2025-го года. В данный период вводилось большинство цифровых технологий в образовательную среду с целью интеграции Запорожской области в общероссийскую систему образования. Определение ключевых моментов внедрения, анализ проблем, с которыми сталкивались образовательные учреждения в данный период времени, который поможет в формировании новой стратегии цифровизации образовательной системы.

Результаты исследования и их обсуждение

Более лояльная и ориентированная на пользователей стратегия внедрения цифровых технологий в сферу образования в Запорожской области с учётом текущей ситуации в регионе. Результат разработки стратегии – пошаговый план по внедрению, представленный в графическом виде.

На рис. 1. представлены основные региональные исполнители во внедрении цифровых технологий и их пользователи.

Главными целями разрабатываемой стратегии являются:

- Лояльность населения к цифровым нововведениям;
- Обеспечение более простого и постепенного внедрения цифровых технологий со стороны конечных исполнителей во внедрении;
- Определение и устранение проблем в процессе внедрения.

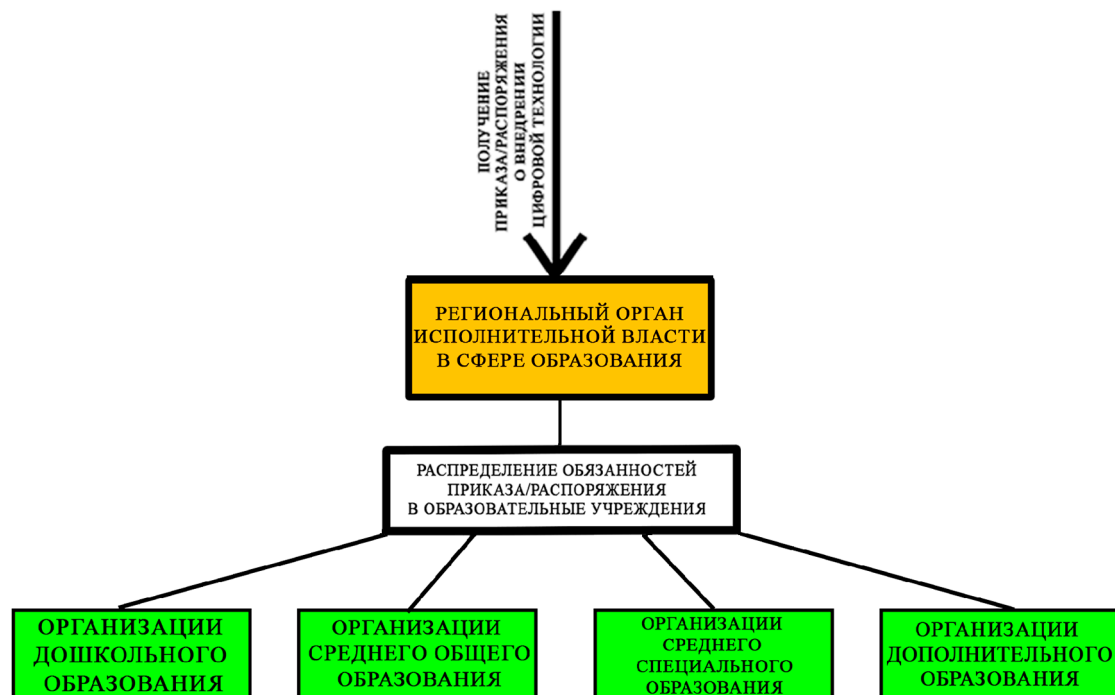


Рис. 1. Основные исполнители во внедрении цифровых технологий и конечные пользователи данных технологий



Рис. 2. Проблемы в цифровизации образования Запорожской области

* Стимулирующие выплаты за внедрение цифровых технологий в образовательное учреждение никаким законом не предусмотрены, т.к. это не относится к оказанию образовательных услуг и не является прямой обязанностью педагогов, которые назначаются ответственными за внедрение той или иной технологии, но в ходе исследования часто фиксировались вопросы педагогов по поводу оплаты труда со стороны рОИВ.

** В следствии острой нехватки кадров, а зачастую отсутствия в штатном расписании должности техника или системного администратора, все работы по внедрению цифровых технологий делегируются на руководителя ОУ*** или руководителем ОУ*** на педагога.

*** ОУ – Образовательное Учреждение.

Исходя из того, что конечным исполнителем Распоряжений и Приказов о внедрении цифровых технологий является не региональный орган исполнительной власти в сфере образования (далее – рОИВ), а образовательные учреждения, подведомственные ему, возникает ряд проблем во внедрении какого-либо новшества, будь то технического и цифрового плана, или любого другого. На рис. 2 описаны все имеющиеся проблемы цифровизации сферы образования в Запорожской области в порядке возрастания критичности и причинно-следственные связи, которые привели к их появлению.

Также к списку проблем можно добавить трудности в взаимодействии рОИВ с ОУ, ведь рОИВ по определённым причинам не может повлиять на ряд проблем, возникающих в ходе внедрения цифровых технологий.

Рассмотрим эти проблемы на примере двух, на данный момент ключевых в цифровой интеграции образования направлений:

1. Создание региональной государственной информационной системы (далее – РГИС), включающая в себя сервисы электронных дневников в связке с сервисом федеральной государственной информационной системы «Моя школа» (далее – ФГИС);

2. Перевод коммуникаций в национальный мессенджер «Макс» в организациях дошкольного, среднего общего, среднего специального и дополнительного образования.

1. Создание региональной государственной информационной системы, включающая в себя сервисы электронных дневников в связке с сервисом федеральной государственной информационной системы «Моя школа»:

Описание: в соответствии с Распоряжением Правительства РФ от 16 июля 2020 г. № 1845-р «Об утверждении методических рекомендаций по порядку формирования и ведения региональных информационных систем, указанных в части 14 статьи 98 Федерального закона «Об образовании в Российской Федерации», в том числе по порядку предоставления родителям (законным представителям) детей сведений из них», статье 98 Федерального закона от 29.12.2012 N

273-ФЗ (ред. от 15.12.2025) «Об образовании в Российской Федерации» [2,3], всем регионам необходимо создать региональные информационные системы в сфере образования для оказания образовательных услуг. Одним из сервисов региональной системы должна быть система электронных журналов и дневников (далее – ЭЖД). В ходе интеграции Запорожской области в общероссийскую систему образования, было также поручено создать свою информационную систему. Параллельно с этой системой, в соответствии с Постановлением Правительства РФ от 13 июля 2022 г. N 1241 [4] также поручено внедрить ФГИС, в котором тоже есть система электронных журналов и дневников.

Проблемы, которые были зафиксированы в ходе исследования:

Крайне сжатые сроки исполнения приказов и проблемы, возникшие при утверждении нормативно-правовых актов для утверждения РГИС на региональном уровне, привели к задержке разработки РГИС, что стало осложнением в дальнейшем внедрении, так как сроки значительно сдвинулись;

Из-за сдвига сроков, работы по внедрению РГИС перенеслись на период с 15 июля 2025 года по 1 сентября 2025 года, по приказу они должны были проводиться с начала мая по 26 июня.

В период с начала июля и до конца августа практически все составы образовательных учреждений был в отпусках, что тоже создало проблему в организации рабочего процесса, в следствии чего сотрудникам образовательных учреждений пришлось работать в период отпуска, что создало напряжение среди работников ОУ.

Помимо организационного момента, были зафиксированы проблемы в районах, приближенных к линии боевого соприкосновения (далее – ЛБС), где часто отсутствовала связь и электроснабжение. Данная проблема есть и на момент написания статьи.

В 60% образовательных учреждений не хватало оборудования для внедрения РГИС и ФГИС.

На табл. 1 обозначена динамика роста внедрения РГИС в ОУ ежемесячно.

Таблица 1

Динамика роста внедрения РГИС в ОУ ежемесячно в %

Июнь	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
0	7	39	70	80	90	95

Таблица 2

Динамика роста внедрения ФГИС в ОУ ежемесячно в %

Август	Сентябрь	Октябрь	Ноябрь	Декабрь
0	40	60	85	100

Таблица 3

Приблизительные данные по внедрению РГИС и ФГИС в других регионах в %
(<Процент по РГИС> / <Процент по ФГИС>)

Месяц	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Название региона						
Запорожская область	7 / 0	39 / 0	70 / 40	80 / 60	90 / 85	95 / 100
Херсонская область	0 / 0	0 / 0	30 / 50	55 / 80	80 / 100	100 / 100
Донецкая Народная Республика	15 / 28	40 / 50	66 / 90	80 / 100	100 / 100	100 / 100
Луганская Народная Республика	0 / 15	0 / 45	10 / 80	55 / 100	100 / 100	100 / 100
Республика Крым	20 / 50	53 / 90	70 / 100	100 / 100	100 / 100	100 / 100
Г. Севастополь	40 / 50	75 / 99	100 / 100	100 / 100	100 / 100	100 / 100

Таблица 4

Приблизительные данные по переводу коммуникаций в национальный мессенджер «Макс» в Запорожской области в период с октября 2025-го года по декабрь 2025-го года в %

Октябрь	Ноябрь	Декабрь
0	13	35

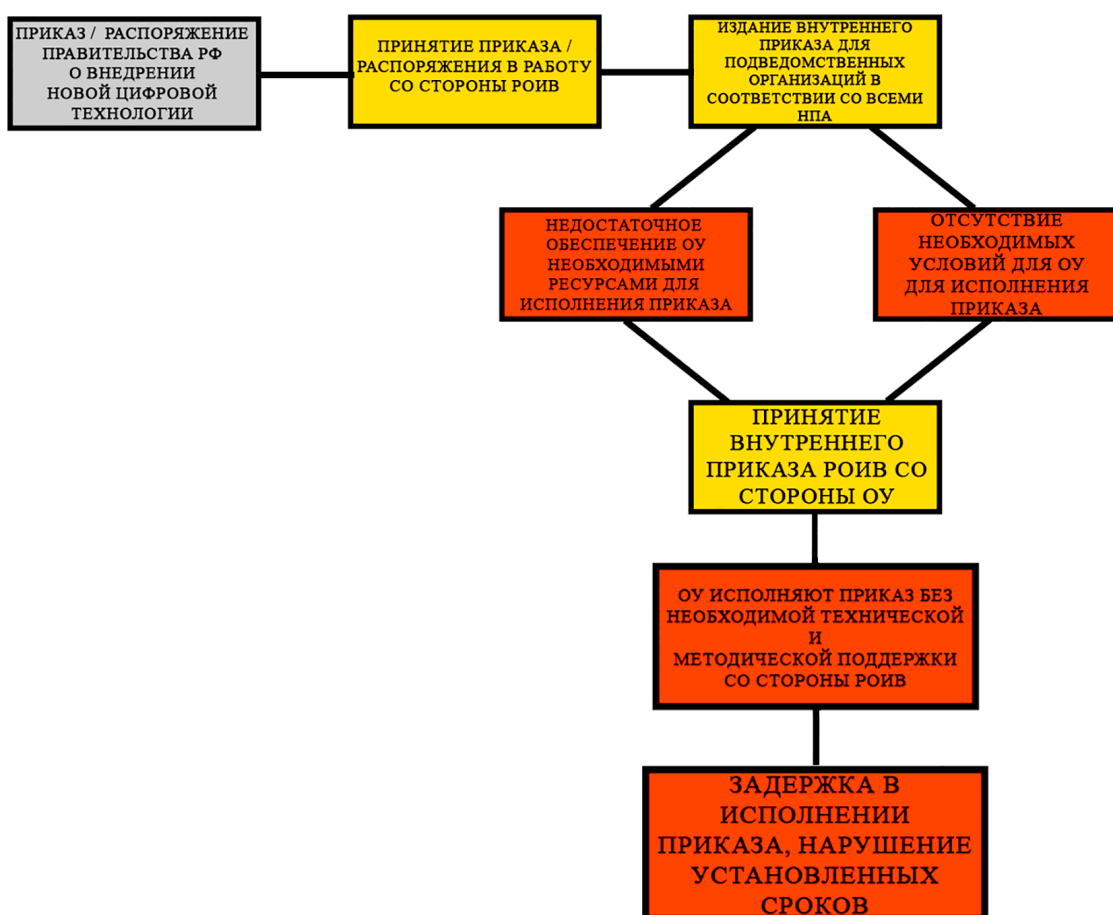


Рис. 3. Графическое изображение текущей модели внедрения цифровых технологий в Запорожской области

ПРЕДВАРИТЕЛЬНЫЕ ДЕЙСТВИЯ СО СТОРОНЫ РОИВ

1. ПОЛНОЦЕННОЕ ТЕХНИЧЕСКОЕ, ТЕХНОЛОГИЧЕСКОЕ И МЕТОДИЧЕСКОЕ СНАБЖЕНИЕ ОУ ДЛЯ УСПЕШНОГО ВНЕДРЕНИЯ ЦИФРОВЫХ ТЕХНОЛОГИЙ.
2. УТВЕРЖДЕНИЕ НОВЫХ ШТАТНЫХ РАСПИСАНИЙ, В КОТОРЫХ БУДУТ ДОЛЖНОСТИ ТЕХНИКА ИЛИ СИСТЕМНОГО АДМИНИСТРАТОРА.
3. ОБЕСПЕЧЕНИЕ БЕСПЕРЕБОЙНОГО И СТАБИЛЬНОГО ЭЛЕКТРОСНАБЖЕНИЯ И ПОДКЛЮЧЕНИЯ К СЕТИ ИНТЕРНЕТ В ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЯХ.
4. НАЛАЖИВАНИЕ ВЗАИМОДЕЙСТВИЯ МЕЖДУ РОИВ И ОУ.

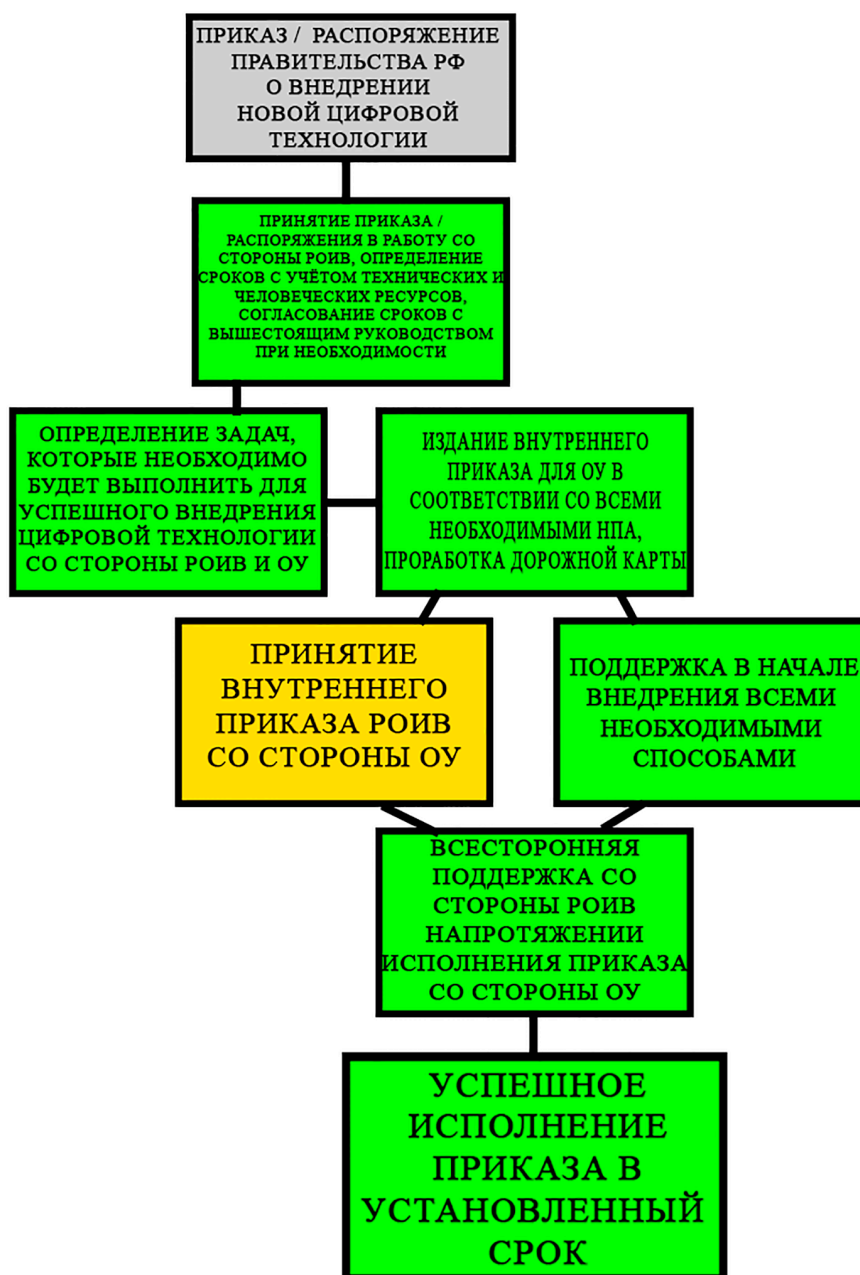


Рис. 4. Новая модель стратегии внедрения цифровых технологий в Запорожской области

Исходя из вышеизложенного можно заметить, что 5% ОУ всё ещё не внедрили РГИС, так как руководство школ категорически против внедрения цифровых нововведений в своей образовательной организации.

Параллельно с внедрением РГИС, в начале августа 2025 года была поставлена задача по внедрению ФГИС. На табл. 2 обозначена динамика роста внедрения ФГИС в ОУ ежемесячно.

Поскольку задач во внедрении ФГИС было меньше, в сравнении с РГИС динамика позитивнее.

После завершения технической части внедрения РГИС и ФГИС, наступил этап организационной работы с педагогами, обучающимися и их родителями, на данном этапе возникло много недовольства со стороны всех вышеперечисленных участников образовательного процесса по причине их недостаточного информирования о системе и её функционале со стороны руководства ОУ, что стало причиной нарушения установленных сроков приказов.

Из-за наличия сразу двух систем с дневниками возникло недопонимание, какую систему использовать, ведь они обе предлагают одинаковый функционал, что также негативно сказалось на мнении пользователей о системах.

Для сравнения результативности, в табл. 3 приведены приблизительные результаты в других новых регионах, а также приблизительные результаты работы в Республике Крым и г. Севастополю.

Исходя из вышеизложенных данных, в других регионах, не смотря на схожую с Запорожской областью ситуацию (за исключением Республики Крым и г. Севастополя), организация рабочего процесса, информирование и техническое снабжение других регионов лучше, чем в Запорожской области и применяются другие стратегии внедрения цифровых технологий, отличающиеся от модели стратегии Запорожской области.

Стоит отметить немаловажный факт того, что в связи с нехваткой кадров в ОУ, рОИВ не может полноценно исполнять свои обязанности как органа, осуществляющего контроль за исполнением поставленных задач, так как в случае нарушения приказа или неподчинения, рОИВ не может санкционировать какие-либо наказания для ОУ, так как острая нехватка кадров не оставляет выбора, кроме как проводить бесчисленные разъяснительные работы с руководством ОУ, нарушившим приказ.

2. Перевод коммуникаций в национальный мессенджер «Макс» в организациях дошкольного, среднего общего, среднего специального и дополнительного образования:

Описание: в соответствии с Федеральным законом №156-ФЗ от 24.06.2025 «О создании многофункционального сервиса обмена информацией и о внесении изменений в отдельные законодательные акты Российской Федерации» [5], ОУ было поручено перевести все рабочие

чаты в национальный мессенджер «Макс», а также создать официальные информационные каналы организаций в срок до 1 января 2026 года.

Проблемы, возникшие при переводе коммуникаций в мессенджер «Макс» в Запорожской области:

Из-за дезинформации о ненадёжности мессенджера, а также оглашения средствах массовой информации, что переход в мессенджер «Макс» проводится на добровольной основе, но поскольку рОИВ была поставлена задача достичь целевого показателя, участники образовательного процесса вынуждены переходить в мессенджер, что уже не являлось добровольным переходом и стало причиной разногласий между рОИВ, выдающим указание, ОУ, исполняющими указание и участниками образовательного процесса – пользователями данного мессенджера. В следствии недостаточного информирования со стороны рОИВ, данная напряжённость на протяжении долгого времени локализовывалась руководством ОУ, результат их работы незначительно улучшил показатель, но не до целевого показателя, установленного приказом.

В табл. 4 приведены приблизительные данные по переводу коммуникаций в национальный мессенджер «Макс» в Запорожской области.

Результаты недостаточного информирования участников образовательного процесса и неправильной стратегии внедрения со стороны рОИВ привели к крайне низкому показателю, далёкому от целевого.

Исходя из вышеизложенного, формируется текущая модель стратегии внедрения цифровых технологий в Запорожской области. Графическое изображение модели представлено на рис. 3.

Исходя из полученного результата анализа, можно увидеть, что множество факторов не учитывается, что негативно сказывается на результатах исполнения приказа. Учитывая прошлые проблемы, формируем новую модель стратегии внедрения цифровых технологий с учётом всех недочётов. Новая модель стратегии внедрения цифровых технологий в Запорожской области представлена в графическом виде на рис. 4.

Заключение

В статье сформирована новая модель стратегии развития и внедрения цифровых технологий в сферу образования Запорожской области, которая облегчит взаимодействие рОИВ с образовательными учреждениями и улучшит результаты цифровизации образовательной среды. Результаты разработки стратегии представляются в графическом виде. Дальнейшее развитие рассмотренной задачи лежит в развитии данной стратегии на уровне рОИВ и тщательной её проработке с образовательными организациями.

Список литературы

1. Распоряжение Правительства РФ от 2 декабря 2021 г. № 3427-р «Об утверждении стратегического направления в области цифровой трансформации образования, относящейся к сфере деятельности Министерства просвещения РФ». URL: <https://www.garant.ru/products/ipo/prime/doc/403075723/> (дата обращения: 15.12.2025).
2. Распоряжение Правительства РФ от 16 июля 2020 г. № 1845-р «Об утверждении методических рекомендаций по порядку формирования и ведения региональных информационных систем, указанных в части 14 статьи 98 Федерального закона «Об образовании в Российской Федерации», в том числе по порядку предоставления родителям (законным представителям) детей сведений из них». URL: <https://www.garant.ru/products/ipo/prime/doc/74299773/> (дата обращения: 15.12.2025).
3. Статья 98. Информационные системы в системе образования Федерального закона от 29.12.2012 N 273-ФЗ (ред. от 15.12.2025) «Об образовании в Российской Федерации». URL: https://www.consultant.ru/document/cons_doc_LAW_140174/07b2a433c35195c28c56a221a970dc3ce333fade/ (дата обращения: 15.12.2025).
4. Постановление Правительства РФ от 13 июля 2022 г. N 1241 «О федеральной государственной информационной системе «Моя школа» и внесении изменения в подпункт «а» пункта 2 Положения об инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг и исполнения государственных и муниципальных функций в электронной форме». URL: <https://base.garant.ru/405000799/> (дата обращения: 15.12.2025).
5. Федеральный закон №156-ФЗ от 24.06.2025 «О создании многофункционального сервиса обмена информацией и о внесении изменений в отдельные законодательные акты Российской Федерации». URL: <https://www.garant.ru/products/ipo/prime/doc/412110426/> (дата обращения: 15.12.2025).

ОБЗОР АЛГОРИТМОВ ДЕТЕКЦИИ ЛИЦ И ИХ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ СИСТЕМЫ УЧЁТА РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКА

Чернышев О.Г.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: a.shapurov@melsu.ru

Научный руководитель: Шануров А. А.

Введение

В эпоху стремительной цифровизации бизнеса и повсеместной автоматизации процессов человеческие ресурсы становятся важнейшим активом любой организации. Эффективность управления персоналом напрямую зависит от точности и объективности данных о его деятельности, а основой для этого служит достоверный учет рабочего времени. Традиционные методы учета, такие как бумажные журналы или электронные таблицы с ручным вводом данных, а также простые системы на основе пропускных карт, не справляются с вызовами современного быстроразвивающегося предприятия. Они подвержены ошибкам, «припискам», «дружескому» учету времени коллегами («buddy punching»), требуют много времени на админи-

стрирование и не обеспечивают необходимого уровня достоверности и аналитики.

В этой связи разработка автоматизированной системы учета рабочего времени, основанной на биометрической идентификации сотрудников с использованием технологий компьютерного зрения, представляется исключительно актуальной и востребованной задачей.

Биометрическая идентификация по лицу – это метод, который, в отличие от пропускной карты, брелока или PIN-кода, невозможно передать, потерять, украсть или подделать без использования сложных высокотехнологичных средств. Это позволяет полностью устранить проблему «buddy punching», когда один сотрудник отмечает приход или уход за другого.

Внедрение такой системы обеспечит справедливые и равные для всех условия учета для всех сотрудников, минимизирует риски внутреннего мошенничества и повышает общий уровень корпоративной безопасности, ограничивая доступ в помещения только авторизованным сотрудникам.

Детекция лиц (Face Detection) является основополагающий и критически важный этап в разработке приложения для учёта рабочего времени сотрудников. Задача алгоритма – автоматическое обнаружение и локализация лиц на изображении или в видеопотоке. Эффективность и точность всей системы распознавания напрямую зависят от качества выполнения этого начального этапа.

Цель исследования – изучить алгоритмы распознавания лиц и провести их сравнительный анализ для создания программного обеспечения, которое будет учитывать рабочее время сотрудников.

Материалы и методы исследования

В процессе исследования были проанализированы научные публикации и техническая документация, которые раскрывают принципы работы, архитектуру, а также преимущества и недостатки трёх основных классов алгоритмов, предназначенных для обнаружения лиц. Анализ носил сравнительный и аналитический характер. В ходе исследования были систематизированы и оценены следующие методы:

Алгоритм Виолы-Джонса (каскады Хаара). Были подробно изучены ключевые составляющие данного алгоритма: признаки Хаара для выявления контрастных особенностей, интегральное изображение для ускорения вычислений и каскадная структура классификаторов для быстрого отбраковки негативных регионов. В качестве практической реализации рассмотрена предобученная модель `haarcascade_frontalface_default.xml` из библиотеки OpenCV.

Метод HOG (Histogram of Oriented Gradients) с классификатором SVM. В этом методе для распознавания объектов используется дескриптор,

который строится на основе гистограмм направлений градиентов яркости в локальных областях изображения. Затем дескриптор нормализуется внутри блоков. Для принятия решений применяется линейный классификатор опорных векторов. В качестве примера реализации этого метода можно привести детектор из библиотеки Dlib.

Методы на основе глубоких нейронных сетей (DNN). Исследованы две основные архитектурные парадигмы: двухэтапные детекторы (например, Faster R-CNN) и одноэтапные детекторы (такие как SSD и YOLO). Особое внимание уделено оптимизированным для работы в реальном времени моделям (например, MobileNet-SSD). В качестве основного инструмента для применения данного класса методов рассмотрен DNN-модуль библиотеки OpenCV (cv2.dnn), который поддерживает загрузку предобученных моделей из фреймворков Caffe, TensorFlow и других. Конкретным примером является легковесная модель `opencv_face_detector_caffe_model`.

Для сравнительного анализа были выбраны следующие критерии:

1. Точность распознавания в различных условиях, включая освещение, ракурс и наличие препятствий.
2. Скорость работы алгоритма.
3. Требования к вычислительным ресурсам, таким как CPU и GPU.
4. Легкость интеграции в уже существующие программные продукты.

Результаты исследования и их обсуждение

Детекция лиц (Face Detection) представляет ключевую задачу компьютерного зрения, заключающаяся в автоматическом обнаружении и локализации лиц на цифровых изображениях

или в видеопотоке. Качество и эффективность этого этапа напрямую влияют на точность всей системы распознавания. В настоящее время для решения этой задачи широко применяются три класса алгоритмов: методы на основе признаков (каскады Хаара), на основе гистограмм ориентированных градиентов и на основе глубоких нейронных сетей. Сравнительный анализ их характеристик является основой для обоснованного выбора технологии для проекта.

1. *Алгоритм Виолы-Джонса (каскады Хаара – Haar Cascades)*. Данный алгоритм, предложенный в 2001 году, стал первым эффективным методом для детекции объектов в реальном времени [1]. Его работа основывается на трёх ключевых идеях (рис. 1):

1. Признаки Хаара (Haar-like Features) – это простые прямоугольные шаблоны, которые вычисляют разность сумм интенсивностей пикселей в белой и чёрной областях. Эти признаки позволяют эффективно выявлять контрастные особенности лица, такие как область глаз (тёмная) относительно переносицы (светлая) или вертикальные границы носа [2, с. 154].

2. Интегральное изображение (Integral Image) – это техника, позволяющая вычислять суммы интенсивностей в прямоугольных областях изображения за константное время, что обеспечивает высокую скорость расчёта большого количества признаков [1].

3. Каскадный классификатор (Cascade of Classifiers) – это алгоритм, основанный на последовательном применении простых классификаторов. Если область изображения не проходит через каждый этап, она сразу исключается из дальнейшего рассмотрения. Благодаря этому основное время тратится только на обработку областей-кандидатов, что обеспечивает высокую скорость работы [1].

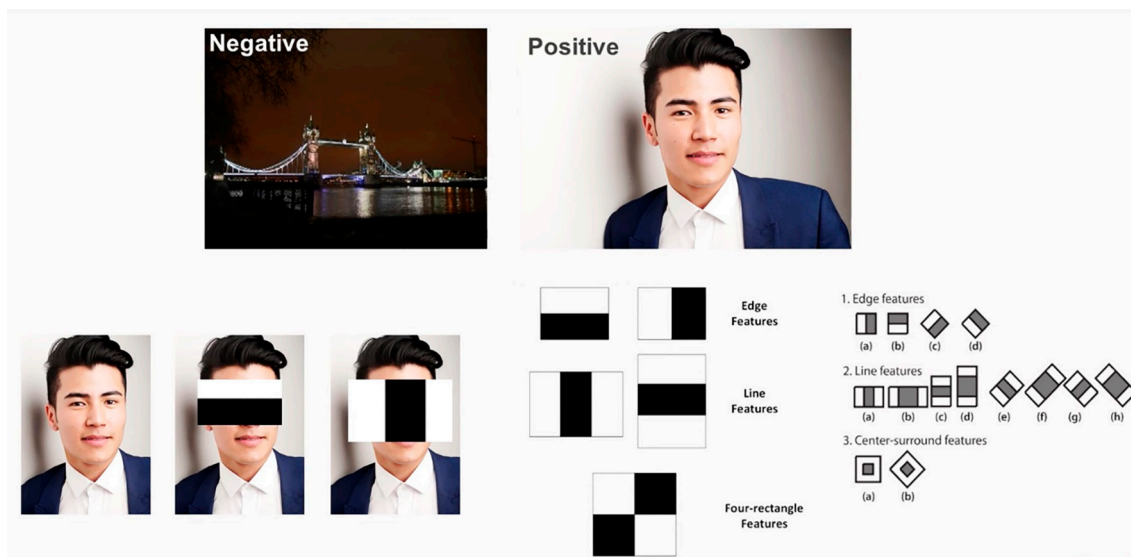


Рис. 1. Алгоритм Виолы-Джонса (каскады Хаара – Haar Cascades) [7]

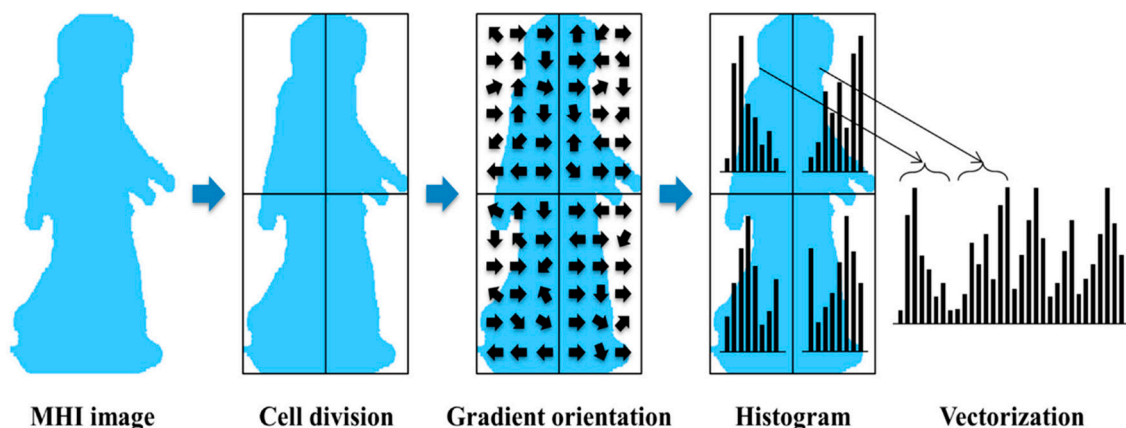


Рис. 2. Метод гистограммы ориентированных градиентов [8]

Преимущества метода включают высокую скорость работы на CPU, простоту реализации и наличие предобученных моделей во многих библиотеках, включая OpenCV (haarcascade_frontalface_default.xml). Основные недостатки – низкая устойчивость к изменениям ракурса (оптимизирован для фронтальных лиц), чувствительность к сложному фону и освещению, а также более низкая точность по сравнению с современными методами на основе глубокого обучения.

2. Метод гистограммы ориентированных градиентов (HOG – Histogram of Oriented Gradients). HOG был первоначально разработан для детекции пешеходов, но успешно адаптирован для лиц в комбинации с классификатором опорных векторов [3, с. 101]. Алгоритм не использует непосредственно интенсивность пикселей, а анализирует структуру объекта (рис. 2).

Принцип работы HOG заключается в следующем: изображение делится на небольшие ячей-

ки, и для каждой из них строится гистограмма направлений градиентов яркости, также известных как ориентации краев. Эти локальные гистограммы затем нормализуются внутри более крупных блоков для компенсации изменений освещённости. Полученный дескриптор HOG описывает форму и контуры объекта [3, с. 103].

Полученный вектор признаков подаётся на вход линейного классификатора SVM, который обучается отличать лицо от не-лица. В сравнении с каскадами Хаара, HOG обычно демонстрирует более высокую точность и лучшую устойчивость к изменению освещения благодаря нормализации градиентов. Однако этот метод, как правило, требует больше вычислительных ресурсов и работает медленнее, чем каскады Хаара, особенно при работе с большими изображениями. Реализация HOG для детекции лиц доступна, например, в библиотеке Dlib.

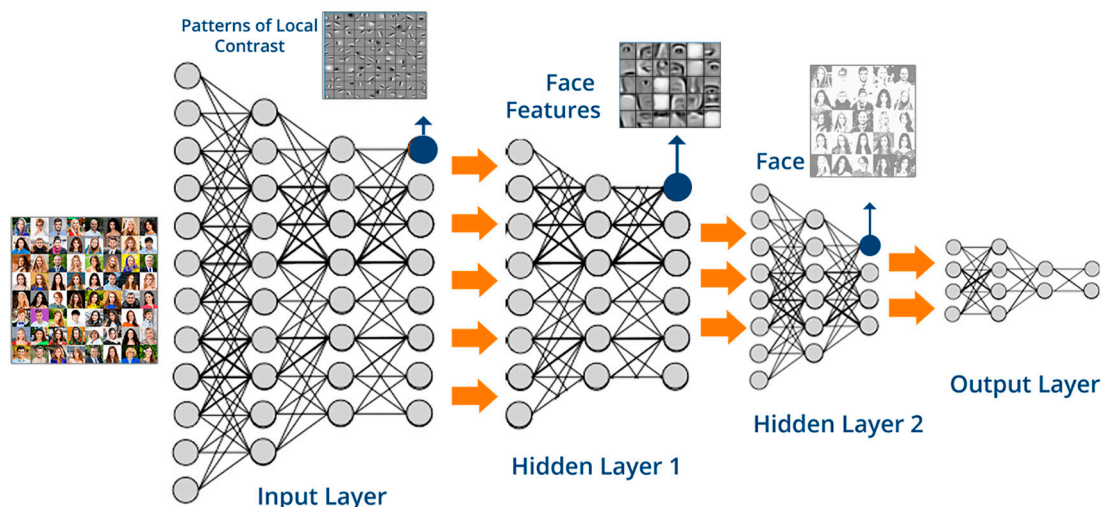


Рис. 3. Deep Neural Networks (Глубокие нейронные сети) [9]

3. Методы на основе глубоких нейронных сетей (DNN – Deep Neural Networks). Современный этап развития детекции лиц связан с применением сверточных нейронных сетей (Convolutional Neural Networks, CNN) и более сложных архитектур [4, с. 17]. Эти методы можно разделить на две основные категории: двухэтапные (two-stage) и одноэтапные (one-stage) детекторы (рис. 3).

Двухэтапные детекторы, такие как Faster R-CNN, работают в два этапа: сначала они выделяют области, которые могут содержать объекты (Region Proposal Network), а затем классифицируют каждый регион и уточняют его границы. Этот подход обеспечивает высокую точность, но требует больше времени для обработки изображений [5, с. 123].

Одноэтапные детекторы (например, SSD – Single Shot MultiBox Detector, YOLO – You Only Look Once): Они прогнозируют координаты ограничивающих рамок (bounding boxes) и вероятности классов непосредственно за один проход по сети. Эти детекторы обеспечивают оптимальный баланс между скоростью и точностью и наиболее пригодны для работы в реальном времени [5, с. 125].

Преимущества DNN-методов неоспоримы: они обеспечивают наивысшую точность детекции, высокую устойчивость к окклюзиям частичному перекрытию лица (окклюзиям), изменению ракурса, масштаба и освещённости. Современные сети обучаются на огромных наборах данных таких как WIDER FACE, что позволяет им обобщать сложные закономерности. Главный недостаток – высокие требования к вычислительным ресурсам. Для обеспечения скорости, приемлемой для реального времени, часто требуется использование графических процессоров (GPU). Тем не менее, существуют оптимизированные легковесные модели такие как MobileNet-SSD, способные работать и на CPU.

При выборе алгоритма распознавания для конкретной системы необходимо учитывать три ключевых параметра: точностью, скоростью и ресурсоёмкостью. Для разрабатываемой системы учёта рабочего времени, где приоритетом является надёжная фиксация событий, наиболее предпочтительным является использование современных методов на основе глубоких нейронных сетей (DNN). Несмотря на повышенные требования к вычислительным ресурсам, ключевые преимущества DNN – высочайшая точность и устойчивость к изменениям ракурса, освещения и наличию аксессуаров – критически важны для обеспечения надёжности системы в реальных, неидеальных условиях офисной среды [4, с. 17].

Использование DNN-модуля библиотеки OpenCV (cv2.dnn) предоставляет уникальную возможность объединить передовую точность

нейросетевых моделей, таких как MobileNet-SSD или OpenPose и удобство интеграции в выбранный технологический стек. Данный модуль поддерживает загрузку предобученных моделей из популярных фреймворков, включая TensorFlow, Caffe, PyTorch, PyCharm, что избавляет от необходимости их сложного обучения с нуля [6]. Для нивелирования повышенной вычислительной нагрузки можно выбрать оптимизированную, легковесную модель например, `opencv_face_detector_caffemodel`, которая поставляется с OpenCV и специально разработана для баланса скорости и качества на CPU. Таким образом, выбор в пользу DNN-детектора в OpenCV является разумным решением для создания надёжной и технологически современной системы, отвечающей строгим требованиям к точности. При условии правильного выбора конкретной архитектуры нейронной сети, DNN-детектор в OpenCV станет основой для создания высокоточного и современного решения.

Выводы

Проведённый сравнительный анализ алгоритмов распознавания лиц наглядно демонстрирует, что методы на основе глубоких нейронных сетей значительно превосходят классические подходы, такие как Haar Cascades, HOG, по ключевому для нашего проекта параметру – точности распознавания в разнообразных условиях. Хотя DNN-методы предъявляют более высокие требования к вычислительным ресурсам, использование оптимизированных предобученных моделей через DNN-модуль OpenCV позволяет успешно применять их на стандартном CPU, что подтверждается практикой внедрения. Таким образом, выбор в пользу DNN-детектора, являющегося стратегическим решением, направленным на обеспечение максимальной надёжности и устойчивости системы биометрического учёта к внешним факторам, таким как освещение, ракурс. Это напрямую влияет на выполнение нефункционального требования по низкому проценту ложных отказов. Классические алгоритмы, как Haar Cascades сохраняют свою актуальность благодаря своей скорости и простоте, однако для решения поставленной задачи с высокими требованиями к достоверности приоритет отдаётся передовой нейросетевой методологии.

Список литературы

1. Viola P., Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). 2001. Vol. 1. URL: <https://ieeexplore.ieee.org/document/990517> (дата обращения: 15.12.2025).
2. Брэдки Г., Келер Дж., Бэхук Э. Освой самостоятельное компьютерное зрение на языке Python за 24 часа / пер. с англ. Москва: Вильямс, 2022. 432 с.
3. Павлов С. С., Федоров А. И. Методы и алгоритмы детектирования лиц в видеопотоке в реальном времени // Известия ЮФУ. Технические науки. 2021. № 5 (214). С. 98–110.

4. Бобина М. А., Волков С. К. Сравнительный анализ алгоритмов распознавания лиц в условиях неидеального освещения // Информационные технологии и вычислительные системы. 2023. № 2. С. 12–24.

5. Шаров В. И., Новикова Т. П. Применение библиотеки OpenCV для задач компьютерного зрения в образовательных проектах // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 1. С. 120–129.

6. Документация библиотеки OpenCV. URL: <https://docs.opencv.org/4.x/> (дата обращения: 15.12.2025).

7. 25 Object Detection with HAAR Cascade Classifiers. URL: <https://www.youtube.com/watch?v=kThRJyQCW-8> (дата обращения: 15.12.2025).

8. Continuous Human Action Recognition Using Depth-MHI-HOG and a Spotter Model. URL: <https://www.mdpi.com/1424-8220/15/3/5197> (дата обращения: 15.12.2025).

9. Deep Learning Vs Machine Learning. URL: <https://k21academy.com/ai-ml/dl-vs-ml/> (дата обращения: 15.12.2025).

ОСОБЕННОСТИ МУЛЬТИМЕДИЙНЫХ ПАРКОВ КАК ОБЪЕКТА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Чикарь Л. А., Мозговенко А. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@amozgovenko.ru*

Задачи исследования:

1. Определить понятие «информационная безопасность».
2. Осуществить анализ существующих типов угроз.
3. Выявить интересы субъектов информационных отношений.
4. Провести анализ мер защиты.

Материалы и методы исследования

Исследования и публикации в области информационной безопасности мультимедийных парков:

1. Научная статья Ирины Владимировны Топчий «Мультимедийные технологии в современном музее как инструмент привлечения посетителей (на примере музея-заповедника «Ар-каим»)» (журнал «Знак: проблемное поле медиаобразования», 2024). В статье рассмотрены возможности использования мультимедийных технологий в пространстве музея, а также классификации таких технологий.

2. Исследование Т. В. Румянцевой «Роль медиапроектов при продвижении музея» (2024). Исследователь отмечает, что оригинальный медиа контент, основанный на уникальной экспозиции, может быть представлен в формате виртуального пространства музея в виде видеоролика или трейлера для популяризации его деятельности [2].

Основная цель – создать защиту информации и поддерживающей её инфраструктуры от случайных или преднамеренных воздействий, которые могут нанести ущерб владельцам или пользователям информации.

Результаты исследования и их обсуждение

Рассмотрим мультимедийный парк как объект информационной безопасности – обеспечение конфиденциальности, целостности и доступности информации в информационных системах, связанных с использованием мультимедийных технологий. Это связано с тем, что мультимедийные данные (изображение, видео, аудио) подвергаются угрозам, и необходимо минимизировать риски, прогнозировать возможные негативные воздействия и предотвращать их.

Виды целей информационной безопасности для мультимедийных парков:

1. Конфиденциальность – доступ к информации только у лиц, имеющих на это полномочия.
2. Целостность – блокировка несанкционированного изменения информации.
3. Доступность – возможность получить требуемую информационную услугу за приемлемое время.
4. Подлинность – полнота и общая точность информации.
5. Неотказуемость – возможность определить источник или авторство информации [1].

Эти цели важны, так как информационные системы создаются для получения определённых информационных услуг, и, если предоставить эти услуги пользователям становится невозможно, это наносит ущерб [3].

Все эти элементы связаны в единую сеть, обеспечивающую бесперебойное функционирование контента и сервисов, и именно эта взаимосвязанность определяет особую уязвимость подобной среды. Любое нарушение в одном из сегментов инфраструктуры может повлечь за собой цепную реакцию сбоев, затронуть критические активы и повлиять на доступность и целостность информации.

С точки зрения информационной безопасности мультимедийные парки относятся к категории объектов с повышенной степенью риска. Это объясняется их публичностью, открытым доступом для большого количества пользователей и разнообразием подключаемых устройств, зачастую использующих стандартные беспроводные интерфейсы и протоколы связи. Высокая плотность цифрового оборудования и использование множества каналов передачи данных (локальные сети, Wi-Fi, Bluetooth, NFC, IT-протоколы) создают широкое поле для потенциальных киберугроз. При этом специфика работы подобных учреждений требует постоянного взаимодействия с внешними источниками информации – облачными платформами, сервисами обновления контента, платёжными шлюзами и базами данных партнёров, что дополнительно увеличивает количество точек возможного воздействия.

Мультимедийные парки одновременно выполняют функции культурно-просветительско-

го учреждения, технологической лаборатории и коммерческого предприятия [7].

Особенности мультимедийных парков как объекта информационной безопасности: Отдельного внимания заслуживает внедрение технологий машинного обучения и искусственного интеллекта в обеспечение информационной безопасности [8]. Алгоритмы анализа поведенческих паттернов позволяют определять аномалии в работе систем и действий пользователей, предсказывать возможные инциденты и минимизировать человеческий фактор при принятии решений. В условиях мультимедийного парка такие технологии позволяют обнаруживать подозрительные действия в реальном времени – например, несанкционированное подключение к локальной сети, попытки доступа к серверу контента или изменения в структуре базы данных посетителей.

*Ключевые угрозы
информационной безопасности*

Атаки на системы управления технологическими процессами (SCADA/OT): взлом систем управления движением, освещением, вентиляцией или механикой аттракциона может привести к созданию опасных ситуаций, причинению вреда здоровью.

Программы-вымогатели (Ransomware): шифрование данных билетных систем, баз данных персонала и гостей, систем управления очередью с целью получения выкупа. Блокировка систем в пиковые часы может парализовать работу всего парка.

Утечки персональных и биометрических данных: кража баз данных посетителей, включая фотографии, платежную информацию, данные о местоположении. Продажа таких данных на черном рынке или использование для целевого фишинга.

1. Атаки типа «отказ в обслуживании» (DDoS): направлены на онлайн-билетные кассы, мобильное приложение парка или внутренние сервисы, что приводит к хаосу на входе и финансовым потерям [6].

2. Компрометация публичных сетей Wi-Fi: организация «зловредных» точек доступа, перехват трафика гостей для кражи учетных данных или данных банковских карт.

3. Физический доступ к оборудованию: незащищенные коммутационные шкафы, медиаплееры или панели управления могут стать точкой входа в сеть для злоумышленника, действующего под видом посетителя.

4. Непреднамеренные воздействия – ошибки пользователя, сбой технических и программных средств информационных систем, природные явления, которые могут привести к искажению, уничтожению, копированию информации.

Для обеспечения информационной безопасности в мультимедийных парках необходимо принимать следующие меры:

1. Технические меры – создание безопасных каналов связи, защита серверов, обеспечение безопасности внешних носителей и рабочих мест пользователей.

2. Организационные меры – подготовка сотрудников, обучение их типовым действиям, необходимым для соблюдения информационной безопасности. Например, проведение тренингов по информационной безопасности, разработка инструкций для сотрудников разных отделов, где поясняются их зоны ответственности и содержатся правила соблюдения режима безопасности. [5]

3. Криптографические методы – шифрование отдельных сообщений и информационного трафика, криптографическая аутентификация объектов сети.

4. Защита протоколов – например, защита физического уровня, обеспечивающего электрические, функциональные и процедурные средства установления, поддержания и разъединения физического соединения, с использованием механизма шифрования.

5. Сегментация сетей (Zero Trust Architecture): обязательное разделение сетей OT, IT, ИТ и гостевого Wi-Fi с помощью межсетевых экранов следующего поколения (NGFW). Принцип «никому не доверяй, проверяй каждый запрос».

6. Всеобъемлющий мониторинг и SIEM-системы: внедрение систем безопасности для операционных технологий (OT Security), сбор и корреляция логов со всех компонентов (датчики, медиасерверы, системы контроля доступа) в едином центре управления информационной безопасностью (SOC).

7. Защита данных и соблюдение нормативных требований: строгое применение принципов Privacy by Design, шифрование данных на rest и in transit, внедрение DLP-систем для предотвращения утечек, регулярные аудиты на соответствие 152-ФЗ и отраслевым стандартам.

8. Регулярное обучение и повышение осведомленности персонала: обязательные кибертренинги для всех сотрудников, от администраторов до аниматоров, с акцентом на фишинг и социальную инженерию. Разработка clear desk и clear screen политик.

9. Инцидент-менеджмент и аварийное восстановление: разработка детальных планов реагирования на инциденты информационной безопасности (кибератака, утечка данных), регулярные учения, наличие актуальных и изолированных резервных копий критических систем.

10. Физическая безопасность информационных технологий -инфраструктуры: ограничение доступа в серверные комнаты и технические помещения, использование средств контроля физического доступа, защита конечных точек (USB-порты и т.д.).

Важно учитывать, что универсальных методов защиты не существует, успех при построении

механизмов безопасности для реальной системы зависит от её индивидуальных особенностей.

Заключение

Мультимедийный парк – это не просто место развлечений, а высокотехнологичный критический объект, где цифровая и физическая безопасность неразделимы. Традиционные подходы к информационной безопасности здесь недостаточны. Требуется специализированная стратегия, учитывающая гибридную природу инфраструктуры, высочайшие требования к доступности и целостности, а также повышенные риски, связанные с обработкой биометрических данных. Упреждающие инвестиции в построение адаптивной, многослойной системы защиты являются не статьей расходов, а ключевым условием устойчивого бизнеса, сохранения репутации и, что самое важное, безопасности тысяч посетителей, доверяющих парку свои данные и досуг.

Список литературы

1. Вострецова Е. В. Основы информационной безопасности: учебное пособие, 2019. 208 с.
2. Буданов Д. Нюансы организации информационной безопасности в музеях // Системы безопасности. 2025. № 2.
3. Богданов А. В., Малыгин И. Г., Синешук Ю. И. Неформальная модель нарушителя безопасности объектов культуры // Вестник Санкт-Петербургского университета ГПС МЧС России. 2013. № 3. С. 109–113. URL: vestnik.igps.ru (дата обращения: 18.10.2025).
4. Корчагин С. И., Павлов В. Г., Бутов А. Н., Ткаченко Д. Г. Подходы к созданию систем обеспечения безопасности особо важных объектов // Системы безопасности. 2010. № 4.
5. Семкин С. Н., Беляков Э. В., Гребенев С. В., Козачок В. И. Основы организационного обеспечения информационной безопасности объектов информатизации: учеб. пособие. М.: Гелиос АРВ, 2005.
6. Климов С. М. Методы и модели противодействия компьютерным атакам. Люберцы: КАТАЛИСТ, 2008.
7. Пресс-конференция о работе мультимедийных исторических парков «Россия – моя история» [Электронный ресурс]. URL: <http://pressmia.ru/pressclub/20171213/951788008.html> (дата обращения: 18.10.2025).
8. Технологии интеллектуального музея нуждаются в этих 7 решениях Интернета вещей // Умный турист. 2022, 27 июня.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОГНОЗИРОВАНИЯ ОБЪЁМА ВЫПУСКА ПРОДУКЦИИ ПРЕДПРИЯТИЯ

Шипилов Д. В., Мозговенко А. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@mozgovenko.ru*

Цель исследования – разработать программное обеспечение на C#, обеспечивающее достоверное прогнозирование объёма выпуска продукции с учётом исторических данных, текущих заказов и производственных ограничений.

Задачи исследования:

1. Проанализировать существующие методы прогнозирования производственных показателей и выбрать оптимальные для реализации.
2. Определить требования к функциональности и интерфейсу ПО с учётом потребностей производственных менеджеров.
3. Разработать архитектуру приложения с модульной структурой (ввод данных, расчёт, визуализация, экспорт).
4. Реализовать алгоритмы обработки временных рядов и статистического анализа на C#.
5. Обеспечить интеграцию с типовыми ERP системами через API или CSV импорт/экспорт.
6. Протестировать ПО на реальных производственных данных и оценить точность прогнозов.

Материалы и методы исследования

Современные исследования в области прогнозирования производственных объёмов фокусируются на:

- Машинном обучении. Публикации 2023–2025 гг. демонстрируют эффективность нейронных сетей (LSTM, GRU) для анализа временных рядов производства. Однако их внедрение требует больших объёмов данных и вычислительных ресурсов.
 - Классических статистических методах. ARIMA, экспоненциальное сглаживание и регрессионный анализ остаются актуальными для предприятий с ограниченной историей данных.
 - Интеграции с IoT. Исследования подчёркивают важность учёта данных с датчиков оборудования для корректировки прогнозов в реальном времени.
 - Облачных решениях. Появление SaaS-платформ для производственного планирования (например, Oracle SCM, SAP IBP) задаёт стандарты юзабилити и масштабируемости.
 - Визуализации данных. Современные работы акцентируют необходимость интерактивных дашбордов с KPI и сценарным анализом.
- Основная цель – создать гибкое, масштабируемое ПО на C#, сочетающее проверенные статистические методы и элементы машинного обучения для прогнозирования объёмов производства.

Результаты исследования и их обсуждение

Для разработки данного программного обеспечения был использован язык программирования C#. Были задействованы следующие библиотеки (рис. 1):

System;
System.Collections.Generic;
System.ComponentModel;
System.Data;
System.Drawing;

System.Linq;
 System.Text;
 System.Threading.Tasks;
 System.Windows.Forms;
 System.IO.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
```

Рис. 1. Библиотеки, которые были использованы при программировании

При загрузке основной формы программы осуществляется загрузка данных по умолчанию, использующая процедуры ReadDataFromCSVFile и ReadWCFromCSVFile (рис. 2).

Если файл WC.CSV отсутствует или пуст, то есть таблица коэффициентов оставшаяся не заполненная, то запускается процедура CalculateWheightcoefficients, которая заполняет таблицу сгенерированными коэффициентами исходя из начальных значений входных данных (рис. 3).

Принудительно запустить процедуру можно, выбрав в меню пункт «Загрузить файл CSV». При этом перед выполнением процедуры откроется окно выбора файла.

Процедуры ReadFromCSVFile и ReadWCFromCSVFiles сначала очищают соответствующие таблицы на форме, а затем считывают текстовые файлы в массив значений и заполняют соответствующие таблицы.

Процедура CalculateWeightCoefficients используется для расчета весовых коэффициентов по умолчанию (рис. 4), перед расчетом таблица коэффициентов очищается в зависимости от начальных значений, берется соответствующее количество строк из таблицы входных данных, высчитывается соотношение к сумме каждого значения, и сортируется по возрастанию. Сумма коэффициентов всегда равна единице, порядок коэффициентов соответствует требуемому соотношению, например $a1 \leq a2 \leq a3$; $A1+a2+a3=1$. Принудительный запуск процедуры заполнения коэффициентов по умолчанию выполняется при нажатии кнопки «Заполнить коэффициенты» (рис. 5).

При нажатии кнопки Рассчитать запускается процедура CalculateMovingAverageMethod. Перед расчетом производится проверка на правильность заполнения коэффициентов, а также проверка заполнения и достаточность входных данных. Далее перебирая каждую строчку таблицы данных производится расчет плановых показателей. По окончании процедуры выполняется метод CalculateSAO, подсчитывающий сумму среднего абсолютного отклонения.

```
string defaultDataFile = Application.StartupPath + "\\data.csv";

if (File.Exists(defaultDataFile))
{
    ReadDataFromCSVFile(defaultDataFile);
    ReadWCfromCSVFile();
    if (dataGridView2.Rows.Count == 0) CalculateWeightCoefficients();
}
}
```

Рис. 2. Код загрузки основной формы по умолчанию

The screenshot shows the 'Forecasts' application window. A 'Файл' (File) menu is open, displaying options: 'Скачать Данные из CSV', 'Сохранить данные в CSV', 'Сохранить расчет в CSV', and 'Выход'. The main window contains a table with the following data:

	Средне (простое)	Разница (простое)	Средне (взвешенное)	Разница (взвешенное)
	0,00	0,00	0,00	0,00
	107,34	49,66	115,57	41,43
	144,00	26,75	140,79	23,54
5 01.07.2020 75	125,42	50,42	128,18	53,18

Рис. 3. Отображение возможных функций работы с файлом

```

Ссылка 1
private void CalculateSAO()
{
    int CountFistRows = Convert.ToInt32(StartCount.Value);
    decimal differentSum = 0;
    for (int i = CountFistRows; i < dataGridView1.Rows.Count-1; i++)
    {
        if (dataGridView1.Rows[i].Cells[0].Value == null) continue;
        if (Decimal.TryParse(dataGridView1.Rows[i].Cells[6].Value.ToString(), out decimal differentValue))
            differentSum += differentValue;
    }
    decimal SAO = differentSum / (dataGridView1.Rows.Count - CountFistRows);
    textBoxSAO.Text = SAO.ToString();
}
    
```

Рис. 4. Отражение весовых коэффициентов

	Коэффициент
a1	0,214285928571429
a2	0,357143214285714
a3	0,428571857142857

Рис. 5. Код расчета САВ

Form2_load. При выборе пункта «Сохранить данные в CSV» (рис. 6).

Выбирая в меню пункт «Сохранить расчет CSV», вызывается окно выбора файла процедурой ExportResultToCSV_click. После успешного выбора вызываются процедуры WriteResultsToCSVFile записывающей результат расчетов в файл и WriteWCToCSVFile для записи таблицы коэффициентов. Таблицы коэффициентов создаются в каталоге программы.

При закрытии программы по умолчанию запускаются процедуры сохранения таблиц формы расчетной и коэффициентов, вызывая процедуры WriteDataToCSVFile и WriteWCToCSVFile.

Заключение

Разработанное ПО для прогнозирования объема выпуска продукции на языке C# демонстрирует высокую практическую ценность для малого и среднего бизнеса, предоставляя доступную альтернативу дорогостоящим облачным решениям.

Список литературы

1. Жданов С. А., Соболева М. Л., Алфимова А. С. Информационные системы: учебник. М.: Прометей, 2015. 302 с.
2. Жиганов С. Н. Анализ динамических систем: учеб. пособие. Саратов: Ай Пи Эр Медиа, 2018. 202 с.
3. Исакова А. И. Предметно-ориентированные экономические информационные системы: учеб. пособие. Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. 238 с.
4. Исаченко О. В. Программное обеспечение компьютерных сетей: учеб. пособие. М.: ИНФРА-М, 2017. 117 с.
5. Ковалева В. Д. Информационные системы в экономике: учеб. пособие. Саратов: Вузовское образование, 2018. 88 с.
6. Коноплева И. А. Информационные системы и технологии управления: учебник. М.: ЮНИТИ-ДАНА, 2017. 591 с.

При нажатии кнопки показать график открывается окно с графиком Form1_load, которое при загрузке формы читает данные из таблицы расчетов данных, строит графики. За это отвечает стандартные процедуры

ЗАКОНЫ ХУДОЖЕСТВЕННОГО ПОВТОРА: ПОЧЕМУ UI-ДИЗАЙН НАСТОЯЩЕГО НАПОМИНАЕТ ПРОШЛОЕ

Якомаскина Т. А., Покуса Т. В.

*ФГБОУ ВО «Мелитопольский государственный
университет», Мелитополь,
e-mail: tana@yakov@yandex.ru*

Научный руководитель: Покуса Т. В.

Введение

В середине 2025 года мобильные интерфейсы демонстрируют смену визуальной концепции, переход происходит от минимализма к глассформизму.

В некоторых случаях резкая смена вызывает неоднозначную реакцию у пользователей, снижается читаемость, появляется визуальный шум и функциональные неудобства.

Возникает вопрос о том, случайно ли это сходство с более ранними визуальными решениями и есть ли в таком случае проявление закона повторяемости художественных циклов. Предполагается, что UI-дизайн подчиняется тем же принципам, что и классическое искусство, но с учетом технологического прогресса.

Цель исследования – выявить существование закономерности цикличности в искусстве как первоисточнике дизайна, сопоставить их с эволюцией интерфейсов и математически обосновать ускорение смены этих циклов.

Материалы и методы исследования

Методологическая основа включала сравнительно-исторический метод для выявления изоморфности искусства и интерфейсов; математическое моделирование для расчёта коэффициента темпоральной компрессии, определяю-

щего скорость смены стилистических парадигм в цифровой и физической среде; спектральный анализ для фиксации частоты циклических колебаний дизайна; анализ данных для построения исследовательских таблиц.

Материалами исследования стали хронологические таблицы, отражающие эволюцию визуальных особенностей: периоды художественных стилей европейского искусства XV–XXI веков и развитие графических интерфейсов.

Результаты исследования и их обсуждение

Анализ исторической динамики художественных стилей показал, что развитие европейского искусства подчиняется некоей логике, описанной как чередование «линейного» и «живописно». Каждая эпоха не возникает спонтанно, это реакция на ограничения и избыточность предыдущей, что формирует собой устойчивый цикл от «Порядка» к «Хаосу» и обратно.

1) Сравнительный анализ динамики визуальной энтропии в классическом искусстве и цифровых интерфейсах

Ренессанс считается начальной точкой современной художественной культуры, закрепив за собой основы гармонии и перспективы. Но к концу XVI века строгость уступает потребности в эмоциональной выразительности и тогда появляется Барокко с динамикой и избыточным декором. Неоклассицизм становится реакцией на перегрузку шумом благодаря использованию строгой геометрии. Но уже в начале XX века Модернизм окончательно отказывается от декоративности используя принцип «форма следует за функцией». А во второй половине века появляется Постмодернизм, вызванный «сенсорным голодом» – эпоха цитирования, иронии и усложнения визуальной формы (табл. 1).

Таблица 1

Эволюция визуальной сложности в классическом искусстве [1]

Художественная эпоха	Временной интервал	Длительность	Визуальная характеристика	Объективная сложность (1 – 10)
Ренессанс (Возрождение)	1420 – 1600	~180 лет	Линейность, статика, симметрия, перспектива	4
Барокко / Рококо	1600 – 1760	~160 лет	Живописность, динамика, избыточный декор, игра теней	9
Неоклассицизм	1760 – 1850	~90 лет	Возврат к античной строгости, чистые линии	3
Модернизм	1900 – 1960	~60 лет	«Форма следует за функцией», отказ от орнамента, геометрия	1
Постмодернизм	1960 – 2000	~40 лет	Сложность, ирония, смешение стилей, коллажность	7
Метамоде́рн	2000 – наст. вр.	~25 лет	Осцилляция между смыслом и формой, новая искренность	5

Таблица 2

Эволюция визуальной сложности в электронных интерфейсах

Этап развития UI	Временной интервал	Длительность	Аналог в искусстве	Описание стиля	Объективная сложность (1 – 10)
Символьный	1980 – 2000	~20 лет	Средневековая графика / Чертеж	Чистый текст, черный фон. Отсутствие метафор. Абстракция.	1
Скевоморфизм	2000 – 2013	~13 лет	Барокко / Реализм	Имитация физических материалов. Статичные тени и блики.	9
Минимализм	2013 – 2020	~7 лет	Баухаус / Супрематизм	Плоские плашки, яркие цвета, отсутствие теней. Векторная графика.	2
Глассформизм	2020 – 2025+	~5 лет	Импрессионизм / Модерн	Полупрозрачность, размытие фона, динамический свет, объем.	7

2) Эмпирическое исследование эволюции визуальных идентификаторов

Эволюция UI-дизайна показывает смену парадигм от «порядка» к «хаосу» и обратно, а главной движущей силой выступает не только эстетика, но и постепенное усиление вычислительных мощностей устройств (табл. 2).

Первый этап – эра символической абстракции, когда единственный доступный интерфейс командная строка или списки. Второй – скевоморфизм – имитация реальных форм, объемов и текстур, которые несут в себе функцию снизить входной порог.

Третий этап – минимализм, когда постепенно исчезают все блики и объемы, значки приобретают символичность, доминируют геометрия и яркие цвета. Четвертый – глассформизм – пространственный дизайн, когда на уже существующий минимализм накладывается новые структура и физика.

Для наглядной демонстрации рассмотрим хронологию изменений графических решений трех ключевых игроков рынка: Google, Microsoft и Apple. Анализ показывает, что несмотря на корпоративные различия, путь развития дизайна подчиняется единому алгоритму.

Изначально логотип Google представляет собой объемную синюю «G» с засечками, окруженную рамкой. С 2008 долго используется строчная «g» – сначала синяя объемная буква на белом фоне. Потом логотип приобретает абстрактность, «g» выходит за пределы квадрата, а объем сохраняется благодаря радиальным градиентам. С 2012 стиль вновь становится строгим, синий сменяется небесно-голубым, буква «возвращается» в рамку, постепенно исчезают тени и блики. В 2015 появляется заглавная «G» с четырьмя основными цветами с четкими границами. В 2025 форма «G» корректируется, а цвета начинают смешиваться на границах благодаря градиенту (рис. 1).

А вот путь, который демонстрирует иконка браузера Chrome от «шара из цветного стекла» с бликами и глубокими тенями она прошла путь к плоской и минималистичной картинке (рис. 2).

Не менее показательна эволюция Microsoft Word, изначально видны попытки вписаться в реализм, но все ещё не хватает разрешения экрана из-за чего рисунок выходит пиксельным, затем мы приходим к пику скевоморфизма и плавно уходим от объема, теней и бликов (рис. 3).



Рис. 1. Эволюция логотипа мобильного приложения Google [2][3]



Рис. 2. Эволюция иконки браузера Chrome [2][3]



Рис. 3. Эволюция иконки «Microsoft Word»[2][3]



Рис. 4. Эволюция иконок iOS1 – iOS26 [2][3]

Наконец система iOS, которая демонстрирует более интересные колебания смены UI-дизайна. С первой по шестую версию преобладал скевоморфизм, который изменился только к седьмой став более минималистичным, но оставляя некоторые фоновые градиенты. И только недавно вышедшая бета-версия системы iOS26 (маркетинговое наименование девятнадцатой версии, которое знаменует выход окончательного варианта в 2026 году) перешла к новой парадигме UI-дизайна – «жидкое стекло», что стало принципиально отличать ее от своих предшественников. Все элементы интерфейса обретают новую физику, многослойность и светотеневые решения, при этом оставляя минималистичные элементы – иконки, к которым уже привыкли (рис. 4).

Анализ визуальной эволюции трехразличных систем подтверждает факт того, что изменения не хаотичны, а имеют определенный порядок от «простого» к «сложному» и наоборот [13, 14].

3) Математическое моделирование темпоральной компрессии и прогнозирование цикличности

Для начала нам нужно доказать, что оба графика имеют одинаковую форму. Необходимо провести расчет коэффициента корреляции Пирсона.

$$r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Для этого мы выбираем три пиковых точки в «искусстве» и «интерфейсах» и составляем из них два вектора сложности:

- вектор $X = \{9; 1; 7\}$

(барокко – модернизм – постмодерн)

- вектор $Y = \{9; 2; 7\}$

(скевоморфизм – минимализм – глассформизм)

Находим среднее арифметическое:

$$\bar{x} = \frac{9+1+7}{3} = \frac{17}{3} \approx 5.67, \bar{y} = \frac{9+2+7}{3} = \frac{18}{3} = 6$$

Считаем отклонения от среднего для каждой точки:

$$X = \{(9 - 5.67 = 3.33); (1 - 5.67 = -4.67); (7 - 5.67 = 1.33)\}$$

$$Y = \{(9 - 6 = 3); (2 - 6 = -4); (7 - 6 = 1)\}$$

Перемножаем отклонения пар и складываем:

$$(3.33 \times 3) + (-4.67 \times -4) + (1.33 \times 1) = 9.99 + 18.68 + 1.33 \approx 30$$

Возводим отклонения в квадрат и суммируем, затем берем корень:

$$\sum (x - \bar{x})^2 = 3.33^2 + (-4.67)^2 + 1.33^2 \approx 11.1 + 21.8 + 1.77 = 34.67$$

$$\Sigma(y - \bar{y})^2 = 3^2 + (-4)^2 + 1^2 \approx 9 + 16 + 1 = 26$$

$$\sqrt{34.67 \times 26} = \sqrt{901.4} \approx 30.02$$

Производим финальный расчет подставив числитель и знаменатель: $r_{xy} \approx 0.999$. Значение почти равно 1. Это значит, что зависимость практически идеальна.

Затем необходимо описать закон сокращения длительности визуальных эпох. Анализ временных рядов показал, что и в искусстве, и в интерфейсах процесс подчиняется закону экспоненциального убывания. Для аппроксимирования использовалась такая функция:

$$T(n) = T_0 \times e^{-\lambda n},$$

где $T(n)$ – длительность n -го цикла, T_0 – начальная длительность, λ – показатель экспоненты сжатия.

В эпохах художественных стилей практически все время встречается одна и та же закономерность – каждый новый срок короче предыдущего вдвое. Поэтому $\lambda_{art} = 0.69$.

В смене интерфейсов нет устоявшейся закономерности:

$$\frac{13}{20} \approx 0.65, \quad \frac{7}{13} \approx 0.54,$$

$$e^{-\lambda} = 0.595, \quad \lambda_{UI} = 0.52$$

Искусство «сжималось» быстрее, а интерфейсы «сжимаются» плавнее, но оба процесса – это экспоненциальный распад времени.

Заключение

Исследование выявило и математически обосновало цикличность развития графических интерфейсов. Сопоставление эволюции художественных стилей XV-XXI веков с историей цифровых интерфейсов подтвердило: дизайн подчиняется универсальным законам смены парадигм, но в ускоренном темпе.

Современные интерфейсы не возвращаются к скевоморфизму, а переходят на новый уровень – «статическая сложность» сменяется «динамической» благодаря снятию аппаратных ограничений. Анализ показал близость стадий трансформации у ключевых компаний.

Текущее усложнение интерфейсов закономерно и прогнозируемо, а дальнейшее развитие UI-дизайна связано с интеграцией AR и стиранием границ между интерфейсом и реальностью.

Список литературы

1. Гомбрих Э. История искусства. М.: Искусство – XXI век, 2013.
2. Иконки приложений [Электронный ресурс]. URL: <https://logos.fandom.com> (дата обращения: 08.12.2025).
3. Иконки приложений [Электронный ресурс]. URL: <https://wikipedia.org> (дата обращения: 08.12.2025).
4. Уолтер А. Эмоциональный веб-дизайн. М.: Манн, Иванов и Фербер, 2012.

5. K. Lee Towards a Working Definition of Designing Generative User Interfaces // DIS '25 Companion, July 5–9, 2025. Funchal, Portugal.

ПРОЦЕСС УПРАВЛЕНИЯ РИСКАМИ И ИНВЕСТИЦИИ В КИБЕРБЕЗОПАСНОСТИ

Ярошенко Е. А., Мозговенко А. А.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,
e-mail: ya@amozgovenko.ru

Задачи исследования:

1. Классифицировать существующие методы оценки экономической эффективности ИБ-инвестиций по критериям:

– тип измеряемых выгод (материальные/нематериальные);

– горизонт планирования;

– требуемые входные данные.

2. Сопоставить методы с этапами процесса управления рисками (идентификация, анализ, реагирование, мониторинг).

Материалы и методы исследования

Современные исследования в области экономики кибербезопасности фокусируются на:

– Квантификации рисков. Работы 2023–2025 гг. развивают модели вероятностной оценки ущерба (FAIR, Cyber Value at Risk), включая каскадные эффекты и репутационные потери. Активно внедряются ML методы для прогнозирования частоты инцидентов.

– ROI метрики для ИБ. Публикуются адаптации классических финансовых показателей (NPV, IRR) под специфику ИБ проектов, где выгоды носят отсроченный и неденежный характер. Обсуждается проблема «тёмной материи» ИБ инвестиций (неучтённые косвенные эффекты).

– Регуляторное влияние. Исследования оценивают затраты на соответствие GDPR, NIS2, ФЗ 152 и др., а также экономический эффект от штрафов за нарушения.

Основная цель – создать методологическую базу для обоснованного распределения инвестиций в кибербезопасность на основе риск-ориентированного подхода.

Результаты исследования и их обсуждение

По данным исследования Gartner кибербезопасность и информационная безопасность занимает первое место в списке запланированных инвестиций на 2022 год, 66% респондентов ожидают увеличения инвестиций в эту сферу [1].

Исследования Accenture свидетельствуют о том, что бюджеты безопасности увеличиваются, более 82% респондентов сказали, что бюджет увеличился в течение 2021 года.

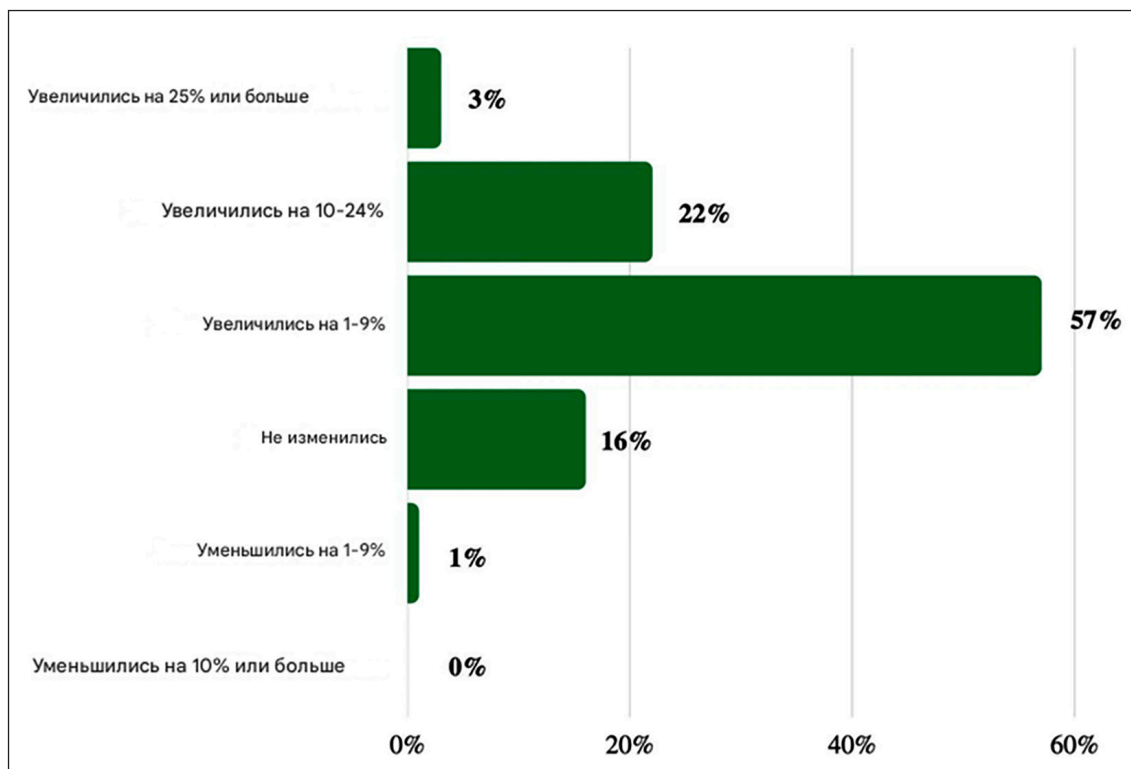


Рис. 1. Увеличение расходов на кибербезопасность в 2021 году по сравнению с 2020 г.

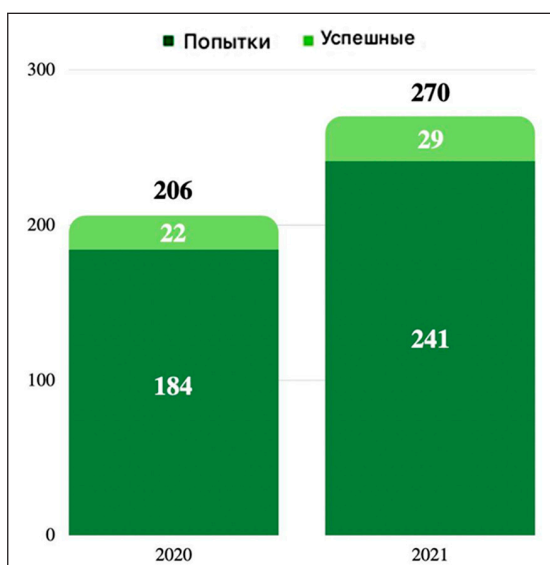


Рис. 2. Увеличение среднего количества атак в 2021 году по сравнению с 2020 г.

Бюджеты на безопасность составляют до 15% от общих расходов на ИТ, что на 5% пунктов больше, чем расходы в 2020 году. На рисунке 1 изображены данные по увеличению расходов на кибербезопасность в 2021 году по сравнению с 2020 годом [2].

По данным Accenture в течение 2021 года на одну компанию было в среднем 270 атак,

что на 31% больше, чем в 2020 году. На рисунке 1.2 изображены данные по увеличению атак в 2021 году по сравнению с 2020 годом [2].

Из-за того, что последствия от определенных атак приводят к значительным репутационным и финансовым ущербом, организация должна инвестировать в кибербезопасность, что позволит избежать определенных киберугроз или потенциально минимизировать последствия успешной реализации киберугроз.

Из-за определенных технических и экономических факторов организации не могут обеспечить совершенный уровень безопасности. Организации обычно планируют ежегодный ограниченный бюджет по кибербезопасности, поэтому перед уполномоченными лицами стоит трудная задача по эффективному распределению данного бюджета. Возникает задача по принятию решения о распределении бюджета по кибербезопасности.

В общем виде процесс принятия решений можно разделить на следующие этапы:

- Характеристика проблемы
- Определение альтернатив
- Определение критериев
- Выбор метода принятия решения
- Оценка альтернатив по критериям
- Проверка решения на соответствие поставленной проблеме [3].

Более подробно рассмотрим каждый из этапов.

Характеристика проблемы

Цель данного этапа – это четкое определение проблемы и описание целей. Лица, принимающие решения, должны согласовать этот вопрос между собой, и кратко и четко описать проблему: определить текущее состояние, цели и требования к решению.

Определение альтернатив

Цель данного этапа – это определение альтернативных решений поставленной проблемы. Каждое решение должно удовлетворять требованиям.

Определение критериев

Цель данного этапа – это выбор критериев, по которым будут оцениваться альтернативные решения. Если количество критериев довольно значительно, то есть смысл объединить их в группы [3].

Выбор метода принятия решения

Существует много способов принятия решения. Выбор метода зависит от поставленной проблемы, количества альтернативных решений и критериев. Также существуют методы, являющие собой сочетание определенных методов.

Оценка альтернатив по критериям

Цель данного этапа – это оценка альтернативных решений по определенным критериям, с помощью выбранного метода.

Проверка решения на соответствие поставленной проблеме

Принятое решение обязательно следует проверять на соответствие поставленным целям и требованиям. Если решение не соответствует, следует пересмотреть альтернативные решения и критерии оценки данных решений.

Существует много методов и подходов, которые используются для оценки инвестиций и принятия решений о распределении бюджета. D. Schatz и R. Bashroush сделали обзор ключевых подходов подробно рассмотрев 25 различных работ по данной тематике, были описаны используемые подходы, их ключевые элементы, преимущества и недостатки.[4]

Также на основе рассмотренных работ были выделены следующие категории методов и подходов к оценке инвестиций:

– ANP (The Analytic Hierarchy Process) – метод анализа иерархий (МАИ), это метод, при котором происходит декомпозиция проблемы: определение цели, критериев оценки и альтернатив, и проводится оценка альтернатив по определенным критериям с помощью попарных сравнений.

– DSS (Decision Support Systems) – системы поддержки принятия решений, это структурированный процесс, помогающий принимать решения более эффективно.

– Game Theory – теория игр, помогающая смоделировать ситуацию (например противостояния «злоумышленник-защитник») и предусмотреть последствия для оптимально-

го принятия решения в условиях конкуренции или конфликта.

– NPV (Net Present Value) – чистая текущая стоимость, используемая для предсказания доходности инвестиции, в общем виде она определяется, как разница стоимости ожидаемых денежных потоков, на сегодня и стоимости инвестиций, на сегодня.

– ROA (Return on Attack) – рентабельность атаки – это расширение ROI, где рассчитывается рентабельность инвестиций злоумышленника, его выгода и расходы.

– ROI (Return on Investment) – рентабельность инвестиций, рассчитывается рентабельность инвестиций, помогающая оценить на сколько выгоден тот или иной денежный вклад.

– ROI, NPV – комбинация рентабельности инвестиций и чистой текущей стоимости.

– ROT (Real Options Theory) – теория реальных опционов, помогающая количественно оценить уровень гибкости, свойственной процессу принятия решений.

– UM (Utility maximization) – максимизация полезности, концепция в которой субъект пытается извлечь наибольшую выгоду из инвестиций [4].

Более подробно рассмотрим некоторые из исследований и работ.

В частности, в работе «Evaluating Information Security Investments Using the ANALYTIC HIERARCHY PROCESS» используется МАИ для оценки инвестиций в информационную безопасность. Они отмечают, что МАИ является полезным инструментом, помогающим организации в принятии инвестиционных решений. Но при использовании данного метода важно и лицо, принимающее решение, поскольку именно он определяет критерии, подкритерии и оценивает [5].

В работе «Measuring the Risk-Based Value of IT Security Solutions» рассматривается подход RROI (Risk-based return on investment) для определения затрат и преимуществ решений по информационной безопасности. Данный метод подходит для определения суммы инвестиций и определяет, что важным критерием при оценке ИТ-решений является польза от инвестиции посредством уменьшения ожидаемых потерь или риска [6].

В работе «An economic modelling approach to information security risk management» представлен подход, позволяющий осуществлять экономическое моделирование процесса управления рисками информационной безопасности. В данном подходе используются оценка рисков, расчет ROSI (Return on Security Investment), NPV и IRR (Internal Rate of Return) [7].

Большое количество работ по теме оценки инвестиций в информационную и кибербезопасность свидетельствует об актуальности дан-

ной проблемы. Среди рассматриваемых работ многие модели для оценки инвестиций базировались на оценке рисков киберугроз. Оценка рисков помогает понять текущие киберугрозы и эффективно распределить инвестиции на улучшение кибербезопасности и минимизацию рисков киберугроз в ИС.

Выводы

Обоснованная оценка инвестиций в кибербезопасность возможна только при тесной связи финансовых метрик с процессами управления рисками. Предложенные методы и рекомендации позволяют:

- повышать прозрачность бюджетирования ИБ;
- аргументировать приоритеты защиты перед руководством;
- адаптировать стратегии под эволюцию угроз.

Дальнейшая работа должна фокусироваться на автоматизации расчётов и создании открытых баз данных по стоимости инцидентов.

Список литературы

1. Gartner Survey of Over 2,000 CIOs Reveals the Need for Enterprises to Embrace Business Composability in 2022, STAMFORD, Conn. Октябрь 18, 2021.
2. Fülöp J. Introduction to decision making methods // BDEI-3 workshop, Washington, 2005. С. 1-15.
3. Schatz D., Bashroush R. Economical valuation for information security investment: systematic literature review // Inf Syst Front 19, 2017. С. 1205-1228.
4. Bodin L. D., Gordon L. A., Loeb M. P. Evaluating information security investments using the analytic hierarchy process // Communications of the ACM. 2005. № 48(2). С. 79-83.
5. Arora A., Hall D., Piato C. A., Ramsey D., Telang R. Measuring the risk-based value of IT security solutions, 2004. P. 35-42.
6. Bojanc R., Jerman-Blažič B. An economic modelling approach to information security risk management // International Journal of Information Management. 2008. № 28(5). С. 413-422.

ПОСТКВАНТОВАЯ КРИПТОГРАФИЯ

Балакина О. А.

*ФГБОУ ВО «Поволжский государственный университет телекоммуникаций и информатики»,
e-mail: olesya_balakina@inbox.ru*

Научный руководитель: Глуценко А. Г.

Введение

Развитие квантовых вычислений является одним из самых значимых технологических про­рывов XXI века. Квантовые компьютеры, основанные на принципах суперпозиции и запутанности, обладают потенциалом решать задачи, недостижимые для классических вычислительных систем. Однако вместе с перспективами, которые открывает квантовая эпоха, человечество сталкивается с новыми угрозами для безопасности информации [2]. Современная криптография, лежащая в основе всех цифровых коммуникаций – от банковских операций до защиты государственных секретов, – строится на предположении о вычислительной сложности определенных математических задач. Если эти задачи могут быть решены эффективно на квантовом компьютере, то привычные алгоритмы шифрования перестают быть надежными.

Именно в этом контексте возникает понятие постквантовой криптографии – области, которая разрабатывает криптографические методы, устойчивые к атакам с использованием квантовых алгоритмов. Постквантовая криптография не отрицает достижения классической криптографии, а, напротив, расширяет их, адаптируя к новой вычислительной парадигме. Это направление стремится обеспечить долгосрочную безопасность цифровых систем в мире, где квантовые компьютеры становятся практической реальностью.

Цель данной статьи – рассмотреть теоретические и прикладные аспекты постквантовой криптографии, описать угрозы, исходящие от квантовых алгоритмов, проанализировать классы криптосистем, которые могут противостоять таким угрозам, и обозначить перспективы развития данного направления в ближай­шие десятилетия.

Квантовые вычисления и угроза современной криптографии

Современные криптографические системы опираются на математические задачи, которые чрезвычайно трудны для решения на классических компьютерах. Так, безопасность широко используемых систем с открытым ключом – RSA, Диффи – Хеллмана и алгоритмов на эллиптических кривых – базируется на сложности факторизации больших целых чисел и вычисления дискретного логарифма. Эти задачи счита-

ются практически нерешаемыми, если речь идет о числах длиной в сотни или тысячи бит.

Квантовые компьютеры радикально меняют ситуацию. В 1994 году Питер Шор предложил квантовый алгоритм, способный эффективно решать задачи факторизации и дискретного логарифмирования. Алгоритм Шора демонстрирует экспоненциальное ускорение по сравнению с классическими методами: время его работы растет полиномиально от числа бит в числе, тогда как классические алгоритмы требуют экспоненциальных затрат. Практическое следствие этого заключается в том, что квантовый компьютер с достаточным числом кубитов сможет за считанные минуты взломать RSA-шифр, на взлом которого классическому суперкомпьютеру понадобились бы миллионы лет.

Для понимания масштабов угрозы рассмотрим пример. Современные стандарты безопасности предполагают использование ключей RSA длиной 2048 бит. В классической модели вычислений это обеспечивает безопасность на уровне, превышающем возможности всех известных суперкомпьютеров. Однако квантовый компьютер, реализующий алгоритм Шора, сможет факторизовать такое число за полиномиальное время, что делает подобные системы бессмысленными в постквантовую эпоху [3]. Аналогичные уязвимости имеют алгоритмы Диффи – Хеллмана и его версия на эллиптических кривых (ECDH), широко применяемые для установления защищенных интернет-соединений (например, в протоколах TLS).

Таким образом, появление масштабируемых квантовых вычислительных систем потенциально разрушает основы современной криптографической инфраструктуры, которая обеспечивает конфиденциальность и аутентификацию в Интернете, мобильных сетях и банковских системах.

Квантовые алгоритмы, влияющие на криптографию

Основными источниками угрозы для криптографии являются два квантовых алгоритма – алгоритм Шора и алгоритм Гровера.

Алгоритм Шора

Как уже отмечалось, алгоритм Шора эффективно решает две ключевые задачи: факторизацию целых чисел и вычисление дискретного логарифма. Его применение делает незащищенными системы, основанные на этих задачах, включая RSA, Diffie–Hellman, ElGamal, а также все схемы на эллиптических кривых (ECC). Это означает, что любые данные, зашифрованные с помощью этих алгоритмов, могут быть расшифрованы, если злоумышленник обладает квантовым компьютером достаточной мощности. Более того, угроза носит ретроспективный

характер: данные, перехваченные сегодня, могут быть дешифрованы в будущем, когда квантовые технологии станут зрелыми. Поэтому переход на постквантовые стандарты – вопрос не только текущей, но и долговременной безопасности.

Алгоритм Гровера

В отличие от алгоритма Шора, алгоритм Гровера не нарушает полностью безопасность симметричных криптосистем, таких как AES или хэш-функций SHA. Однако он обеспечивает квадратичное ускорение при поиске ключа методом перебора: если для классического компьютера требуется порядка 2^n операций для перебора ключа длиной n бит, то для квантового компьютера – лишь $2^{n/2}$. Следовательно, уровень защиты симметричных систем фактически уменьшается вдвое. Например, AES-256 при атаке с использованием алгоритма Гровера обеспечивает ту же устойчивость, что и AES-128 при классической атаке. Это означает, что для сохранения эквивалентного уровня безопасности необходимо удвоить длину ключа [3].

Основы постквантовой криптографии

Постквантовая криптография – это область, разрабатывающая криптографические системы, устойчивые к атакам с использованием квантовых алгоритмов. Главное отличие ПКК от квантовой криптографии состоит в том, что постквантовые алгоритмы реализуются на классических вычислительных устройствах, не требуя квантовых каналов связи [1]. Они должны быть совместимы с существующей инфраструктурой, но при этом обеспечивать безопасность даже в условиях появления квантовых компьютеров.

Основная задача исследователей заключается в поиске новых математических проблем, для которых не существует эффективных квантовых алгоритмов. Эти задачи должны удовлетворять следующим критериям:

1. Трудность решения как на классическом, так и на квантовом компьютере;
2. Возможность практической реализации криптографических протоколов;
3. Эффективность по скорости и объему ключей;
4. Доказуемая безопасность на основе известных сложных задач.

На сегодняшний день выделяют несколько основных направлений постквантовой криптографии [1]:

1. Решеточная криптография

Одним из наиболее перспективных направлений является криптография на основе решеток. Она базируется на математических задачах, связанных с поиском кратчайшего вектора в решетке (SVP) и задачей ближайшего вектора (CVP). Эти задачи остаются сложными как для классических, так и для квантовых компьютеров, что делает их надежной основой для криптосистем нового поколения.

Решеточные схемы обладают рядом преимуществ:

- доказуемая безопасность, основанная на NP-трудных задачах;
- высокая производительность;
- простота реализации на классических процессорах;
- возможность построения не только шифрования, но и цифровых подписей, а также функций хэширования и анонимных идентификаторов.

Классическим примером решеточных схем является криптосистема NTRU, предложенная в 1996 году. Она использует арифметику в кольцах многочленов и обеспечивает высокую скорость шифрования и дешифрования при малом размере ключей. Позже появились более универсальные конструкции – схемы, основанные на LWE (Learning with errors) и ее вариант Ring-LWE, которые легли в основу множества современных постквантовых алгоритмов, включая Kyber и Dilithium [4].

Решеточные схемы особенно привлекательны для стандартизации, поскольку они обладают сбалансированными характеристиками безопасности и эффективности. Именно поэтому в рамках инициативы NIST PQC (Национального института стандартов и технологий США) именно решеточные алгоритмы были выбраны в качестве основных кандидатов для стандартизации постквантовых протоколов.

2. Кодовая криптография

Другим исторически значимым направлением является криптография на основе теории кодирования. Ее основой служит задача декодирования случайных линейных кодов – математически трудная задача, не имеющая известных полиномиальных решений даже при использовании квантовых алгоритмов. Первой практической реализацией стала криптосистема МакЭлиса, предложенная в 1978 году. Она основана на использовании кодов Гоппы и позволяет реализовать стойкое шифрование.

Главным преимуществом кодовых криптосистем является высокая устойчивость к известным видам атак. Недостаток – большие размеры открытых и закрытых ключей (порядка сотен килобайт и более), что затрудняет применение в устройствах с ограниченными ресурсами. Тем не менее, для некоторых задач, таких как защита критически важных данных и квантово-устойчивая передача ключей, эти схемы остаются актуальными.

3. Мультивариантная криптография

Мультивариантные криптосистемы основаны на трудности решения систем многочленов над конечными полями. В общем виде задача заключается в нахождении набора переменных, удовлетворяющих множеству нелинейных уравнений. С математической точки зрения это NP-трудная задача, что делает ее привлекательной основой для построения криптографических схем.

Исторически первым известным примером является система MI (Matsumoto–Imai), предложенная в конце 1980-х годов. Позднее появились улучшенные схемы, такие как HFE (Hidden Field Equations) и ее модификации, включая Rainbow – одну из наиболее известных и исследованных мультивариантных схем цифровой подписи.

Преимуществом мультивариантной криптографии является высокая скорость выполнения операций, особенно на этапе верификации подписи. Однако существенным недостатком долгое время считались большие размеры открытых ключей (часто десятки и сотни килобайт), а также неоднократные компрометации некоторых предложенных схем вследствие алгебраических атак. Несмотря на это, направление остается активно развивающимся, и новые подходы к генерации уравнений позволяют повышать устойчивость систем без значительного увеличения вычислительной нагрузки.

Особый интерес представляют гибридные схемы, в которых мультивариантные подписи комбинируются с решеточными или кодовыми конструкциями. Такой подход позволяет достичь как высокой производительности, так и теоретической стойкости.

4. Криптография на основе изогений эллиптических кривых

Еще одним оригинальным направлением постквантовой криптографии является использование изогений эллиптических кривых. Изогения – это морфизмы между эллиптическими кривыми, сохраняющие их структуру. Безопасность таких систем основана на сложности нахождения изогении между двумя заданными кривыми, что представляет собой математическую задачу, для которой пока не найдено эффективного квантового алгоритма.

Наиболее известной схемой этого типа является SIDH (Supersingular Isogeny Diffie–Hellman), а ее оптимизированная версия – SIKE (Supersingular Isogeny Key Encapsulation) – долгое время считалась одним из самых перспективных кандидатов на стандартизацию. Изогенные криптосистемы привлекательны тем, что обеспечивают чрезвычайно компактные ключи (всего несколько сотен байт), что делает их особенно удобными для применения в мобильных устройствах и встроенных системах.

Однако в 2022 году стало известно, что схема SIKE подвержена атакам, основанным на анализе структуры используемых кривых и вычислении специальных подгрупп. Это показало, что изогенная криптография еще не достигла зрелости, необходимой для массового внедрения, но направление остается важным объектом фундаментальных исследований.

В долгосрочной перспективе изогенные методы могут стать основой для легких и энерго-

эффективных постквантовых протоколов, особенно если удастся создать универсальные решения с доказуемыми гарантиями безопасности.

5. Криптография на основе хэш-функций

Хэш-ориентированные криптографические схемы считаются одними из самых надежных и проверенных решений постквантового периода. Их безопасность опирается исключительно на свойства криптографических хэш-функций, а именно на устойчивость к коллизиям и предобразам. Поскольку хэширование является базовой операцией, к которой пока не найдено эффективных квантовых алгоритмов (кроме алгоритма Гровера, снижающего безопасность лишь вдвое), хэш-ориентированные схемы обладают высокой устойчивостью к квантовым атакам.

Наиболее известными представителями этого направления являются схемы цифровой подписи: Lamport, Merkle и их современные варианты – XMSS (eXtended Merkle Signature Scheme) и SPHINCS+.

- Lamport Signature – простейшая схема, использующая одноразовые ключи, но требующая большого количества данных;

- Merkle Signature – развивает идею одноразовых подписей, объединяя их в древовидную структуру, что позволяет многократное использование публичного ключа;

- XMSS и SPHINCS+ – современные стандартизированные схемы, обеспечивающие высокую степень защиты и практичность применения.

Главный недостаток хэш-подписей – значительные размеры ключей и относительно медленная генерация подписи. Тем не менее их устойчивость к квантовым атакам делает их незаменимыми в ситуациях, где необходима долговременная сохранность данных, например, при хранении государственных архивов, юридически значимых документов и медицинских записей.

Стандартизация постквантовой криптографии

Понимая надвигающуюся угрозу, международное сообщество активно работает над созданием единых стандартов постквантовой криптографии. Ключевым центром этих инициатив является Национальный институт стандартов и технологий США (NIST), который в 2016 году запустил многоэтапный проект по отбору и оценке постквантовых алгоритмов.

Цель программы NIST PQC – выбрать набор алгоритмов, которые смогут заменить современные схемы RSA и ECC в протоколах TLS, IPsec, SSH и других. Критериями отбора стали:

- математическая устойчивость к известным классическим и квантовым атакам;
- эффективность реализации на различных аппаратных и программных платформах;

- устойчивость к побочным каналам;
- гибкость параметров и совместимость с существующими протоколами.

После нескольких раундов анализа, включающих более 80 предложенных схем, в июле 2022 года NIST объявил первые алгоритмы, выбранные для стандартизации:

- CRYSTALS-Kyber – схема шифрования и обмена ключами, основанная на решеточных задачах LWE;
- CRYSTALS-Dilithium – схема цифровой подписи на основе решеточных конструкций;
- Falcon – более компактная решеточная схема подписи;
- SPHINCS+ – хэш-ориентированная подпись с долгосрочной безопасностью.

Эти алгоритмы демонстрируют оптимальное сочетание производительности и устойчивости, что позволяет постепенно интегрировать их в современные коммуникационные стандарты. В ближайшие годы ожидается появление финальных версий стандартов, после чего начнется глобальный процесс миграции инфраструктуры безопасности [4].

Гибридные подходы и переходный период

Переход от классических к постквантовым системам не может быть мгновенным. Большинство современных сетевых протоколов, включая HTTPS, VPN и почтовые сервисы, тесно связаны с инфраструктурой открытых ключей (PKI), основанной на RSA и ECC. Резкая замена алгоритмов может привести к несовместимости и сбоям в работе систем.

Для решения этой проблемы разработчики используют гибридные криптографические схемы, которые объединяют классические и постквантовые алгоритмы. Например, при установлении защищенного соединения может использоваться комбинация ECDH и Kyber: даже если одна из схем будет скомпрометирована, вторая обеспечит защиту.

Такой подход имеет следующие преимущества:

- плавный переход к новой криптографии без нарушения совместимости;
- возможность постепенного тестирования и оптимизации новых алгоритмов;
- дополнительная устойчивость к непредвиденным атакам на новые схемы.

Параллельно с техническими мерами важную роль играет образование специалистов и обновление нормативной базы. Переход к постквантовым стандартам требует перепроектирования систем сертификации, инфраструктуры ключей и аппаратных модулей безопасности. Все это представляет масштабную задачу для государственных структур, бизнеса и академического сообщества [2].

Практические применения и вызовы внедрения

Несмотря на активные исследования, внедрение постквантовой криптографии в практику сопряжено с рядом сложностей. Основные из них:

1. Размер ключей и сообщений. Некоторые схемы, особенно кодовые и мультивариантные, требуют больших объемов памяти, что затрудняет их использование в IoT-устройствах;

2. Скорость вычислений. Хотя решеточные схемы достаточно быстры, в некоторых случаях они уступают по эффективности классическим алгоритмам, что может замедлить работу серверов;

3. Совместимость протоколов. Новые алгоритмы необходимо адаптировать к существующим стандартам коммуникаций (TLS, SSH, IPsec и т. д.), что требует глубокой переработки инфраструктуры;

4. Безопасность на практике. Реальные реализации подвержены атакам через побочные каналы – измерение времени, потребления энергии и электромагнитных излучений.

Тем не менее ряд компаний уже начал интеграцию постквантовых решений. Google тестировал гибридные алгоритмы в протоколе TLS 1.3; Microsoft и Cloudflare проводят пилотные проекты с Kyber и Dilithium; а правительственные структуры США и ЕС разрабатывают национальные дорожные карты перехода.

Особое внимание уделяется критическим секторам – энергетике, телекоммуникациям и обороне, где долгосрочная сохранность секретных данных имеет первостепенное значение. Даже если квантовые компьютеры массово появятся лишь через 10–15 лет, данные, зашифрованные сегодня, должны оставаться защищенными в будущем.

Перспективы развития постквантовой криптографии

Постквантовая криптография находится на этапе активного становления. На горизонте ближайших десятилетий можно выделить несколько ключевых тенденций ее развития:

1. Интеграция в глобальные стандарты. После окончательной стандартизации NIST и ISO/IEC начнется широкое внедрение PQ-алгоритмов во все уровни сетевых протоколов;

2. Оптимизация и миниатюризация. Активно ведутся работы по сокращению размеров ключей и повышению скорости работы схем, что позволит применять их даже в микроконтроллерах;

3. Комбинированные архитектуры. Гибридные схемы, объединяющие разные математические подходы (решетки + хэши + изогении), будут играть ключевую роль в обеспечении надежности;

4. Развитие формальных доказательств безопасности. Все больше внимания уделяется

строгим математическим доказательствам, связывающим стойкость алгоритмов с фундаментальными задачами теории сложности;

5. Создание национальных стратегий киберустойчивости. Государства разрабатывают программы подготовки инфраструктуры к постквантовой эпохе, включая сертификацию оборудования и обучение специалистов.

Заключение

Постквантовая криптография – это ответ человечества на технологический вызов, который несут квантовые вычисления. В течение десятилетий классические криптографические методы считались незыблемыми, но появление алгоритмов Шора и Гровера продемонстрировало: безопасность, основанная только на вычислительной сложности, не является вечной.

Разработка и внедрение криптографических схем, устойчивых к квантовым атакам, – ключевая задача для государств, корпораций и исследовательских центров [2]. Решеточные, кодовые, хэш-ориентированные и мультивариантные системы представляют собой фундамент, на котором будет построена новая инфраструктура доверия. Уже сегодня стандарты NIST PQС формируют основу для перехода к безопасным коммуникациям в постквантовую эпоху.

Однако технологический прогресс требует не только новых алгоритмов, но и изменения

мышления. Необходимо осознать, что защита данных – это процесс, а не состояние. Постквантовая криптография должна стать не разовой заменой старых стандартов, а частью динамичной, постоянно обновляемой системы глобальной безопасности.

В обозримом будущем квантовые компьютеры, вероятно, станут столь же повседневными, как современные серверы. И когда это произойдет, устойчивость цифрового мира будет зависеть от того, насколько успешно сегодня человечество создаст фундамент постквантовой безопасности – надежный, прозрачный и совместимый с принципами открытого интернета.

Список литературы

1. URL: [https://ru.wikipedia.org/wiki/Постквантовая криптография](https://ru.wikipedia.org/wiki/Постквантовая_криптография) [Электронный ресурс]. (дата обращения: 15.12.2025).
2. Маршев И. С. Постквантовая Криптография // Материалы XVII Международной студенческой научной конференции «Студенческий научный форум – 2025» [Электронный ресурс]. URL: <https://files.scienceforum.ru/pdf/2025/6746d1ffaed12.pdf> (дата обращения: 15.12.2025).
3. Букашкин С. А., Черепнев М. А. Квантовый компьютер и постквантовая криптография // Программная инженерия. 2021. №4. С. 171-178. [Электронный ресурс]. URL: <http://novtex.ru/prin/rus/10.17587/prin.12.171-178.html> (дата обращения: 15.12.2025).
4. Федоров С. К. Анализ постквантовых криптографических алгоритмов // Аллея Науки. 2021. №6 (57). [Электронный ресурс]. URL: https://alley-science.ru/domains_data/files/1June2021/ANALIZ%20POSTKvantovyn%20KRIPTOGRAFICHESKIH%20ALGORITMOV.pdf (дата обращения: 15.12.2025).

*Экономические науки***ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И МЕТОДЫ ОПТИМИЗАЦИИ
В ЦИФРОВОЙ ЭКОНОМИКЕ**

Мельник Е. Г.

*ФГБОУ ВО «Мелитопольский
государственный университет»,
Мелитополь, e-mail: elena27712@mail.ru**Научный руководитель: Покуса Т. В.***Введение**

В настоящее время информационные технологии являются одним из ключевых факторов развития общества и экономики. Их внедрение влияет на управление, промышленность, образование, медицину и бизнес, обеспечивая автоматизацию процессов, повышение скорости обработки данных и улучшение качества услуг.

При этом цифровая трансформация сопровождается ростом роли данных и цифровой инфраструктуры, а также расширением применения аналитических и интеллектуальных инструментов в организациях [1].

Современный этап развития ИТ характеризуется быстрым обновлением программных и аппаратных решений, переходом к цифровым платформам и активным внедрением технологий анализа данных. Эти процессы формируют новые требования к информационным системам и компетенциям специалистов, а также повышают актуальность исследований, связанных с тенденциями развития ИТ и их практическими эффектами [1].

Целью исследования является анализ современных тенденций развития информационных технологий и определение их роли в условиях цифровой трансформации общества и экономики.

Материалы и методы исследования

В работе использованы методы анализа, сравнения и систематизации научной информации. Информационной базой послужили научные публикации и официальные материалы,

посвящённые цифровым технологиям и их применению в экономике и управлении [1].

**Результаты исследования
и их обсуждение**

Анализ источников показывает, что развитие информационных технологий на современном этапе определяется комплексом взаимосвязанных тенденций: цифровизацией процессов, расширением облачной инфраструктуры, внедрением интеллектуальных методов обработки данных, а также усилением требований к защите информации (таблица 1) [1].

Цифровизация предполагает перевод данных, услуг и бизнес-процессов в электронный вид и развитие цифровой инфраструктуры организаций. В результате повышается управляемость процессов, ускоряется документооборот и упрощается взаимодействие между подразделениями и внешними контрагентами [1].

Облачные технологии стали базовым инфраструктурным решением для хранения и обработки данных и предоставления сервисов. Переход к модели облачных вычислений связывают с гибкостью масштабирования, снижением затрат на поддержку локальной инфраструктуры и удобством доступа к ресурсам. При этом в научной и практической литературе широко используется определение облачных вычислений как модели, обеспечивающей удобный сетевой доступ «по требованию» к общему пулу вычислительных ресурсов [3]. Дополнительно отмечается, что облачные платформы формируют инфраструктурный уровень цифровой трансформации, на базе которого развиваются API-интеграции и сервисные экосистемы [2].

Искусственный интеллект и аналитика данных (big data) применяются для прогнозирования, классификации, выявления закономерностей и поддержки принятия решений. В экономике такие подходы используются, например, в кредитном скоринге, управлении рисками и клиентской аналитике, а рост объёмов данных делает применение методов big data и машинного обучения практически необходимым [2].

Таблица 1

Основные тенденции развития информационных технологий

№	Тенденция	Характеристика
1	Цифровизация	Переход процессов и услуг в цифровую форму
2	Облачные технологии	Использование удаленных вычислительных ресурсов
3	Искусственный интеллект	Автоматизация анализа данных и принятия решений
4	Большие данные	Обработка и анализ больших объемов информации
5	Информационная безопасность	Защита данных и информационных систем

Таблица 2

Применение информационных технологий в различных сферах

Сфера	Применение ИТ
Экономика	Автоматизация учета и управления
Образование	Дистанционное обучение, электронные ресурсы
Медицина	Медицинские информационные системы
Государственное управление	Электронные государственные услуги
Бизнес	CRM-системы, аналитические платформы

Одновременно с цифровизацией возрастает значимость *информационной безопасности*, так как расширение цифровых каналов и рост массивов данных увеличивают потенциальные риски утечек и несанкционированного доступа. В этих условиях защита информации рассматривается как обязательный элемент устойчивого развития цифровой экономики и функционирования информационных систем [4].

Рассмотренные тенденции находят практическое применение в различных сферах деятельности (таблица 2).

В частности, в экономике и управлении ИТ используются не только для автоматизации учёта и анализа, но и для реализации методов *математического моделирования и оптимизации*, позволяющих обоснованно выбирать управленческие решения при ограниченных ресурсах [5].

Пример применения методов оптимизации (линейное программирование, симплекс-метод)

Рассмотрим типичную экономическую задачу планирования выпуска продукции. Пусть предприятие производит два вида продукции: x_1 и x_2 . Прибыль от единицы продукции составляет 40 и 30 усл. ед. соответственно. Производство ограничено ресурсами:

ресурс А: не более 100 ед.; расход $2x_1 + 1x_2$;
ресурс В: не более 80 ед.; расход $1x_1 + 2x_2$.

Тогда математическая модель задачи имеет вид:

$$\max Z = 40x_1 + 30x_2$$

$$\begin{cases} 2x_1 + x_2 \leq 100, \\ x_1 + 2x_2 \leq 80, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Данная модель относится к задачам *линейного программирования* и может быть решена

симплекс-методом. Оптимальный план соответствует $x_1 = 40, x_2 = 20$, при этом максимальная прибыль $Z = 2200$ усл. ед. [5].

Использование программных средств в составе экономических информационных систем позволяет автоматизировать подобные расчёты и применять оптимизационные методы в планировании и анализе [5].

Заключение

Проведённый анализ показывает, что современные тенденции развития информационных технологий связаны с цифровизацией, развитием облачной инфраструктуры, внедрением методов анализа данных и искусственного интеллекта, а также усилением требований к информационной безопасности [1–4]. Дополнительно показано, что в экономике ИТ обеспечивают практическое применение математического моделирования и оптимизационных методов (в том числе симплекс-метода), повышая обоснованность управленческих решений [5].

Список литературы

1. Морковкин Д. Е. Анализ использования цифровых технологий в сфере научных исследований и разработок в контексте обеспечения промышленного суверенитета российской экономики // Вестник Евразийской науки. 2024. Т. 16. № 6. URL: <https://esj.today/PDF/30FAVN624.pdf> (дата обращения: 15.12.2025).
2. Милич А. Д. Цифровые технологии как фактор трансформации финансового сектора // Вестник Евразийской науки. 2025. Т. 17. № 5. URL: <https://esj.today/PDF/46ECVN525.pdf> (дата обращения: 15.12.2025).
3. NIST Cloud Computing Standards Roadmap (SP 500-291r2). National Institute of Standards and Technology, 2013. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pdf> (дата обращения: 15.12.2025).
4. ENISA Threat Landscape 2024 [Электронный ресурс]. European Union Agency for Cybersecurity (ENISA). September 2024. URL: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024> (дата обращения: 15.12.2025).
5. Муравьева Е. В. Линейное программирование: методические указания к практическим занятиям. Челябинск: Издательский центр ЮУрГУ, 2012. 44 с.

Евразийская научно-промышленная палата

МАТЕРИАЛЫ МЕЖДУНАРОДНОЙ
СТУДЕНЧЕСКОЙ НАУЧНОЙ КОНФЕРЕНЦИИ
«СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ»

ТОМ XIX



Компьютерная верстка Доронкиной Е.Н.

Сдано в набор 10.03.2026
Подписано в печать 14.04.2026
Формат 60×90¹/₈. Бумага офсетная 80 г/м²
Гарнитура «Times New Roman»
Печать цифровая. Усл. печ. л. 17
Тираж 500 экз. Заказ ОК 2026/XIX

Отпечатано в типографии Евразийской научно-промышленной палаты,
410035, г. Саратов, ул. Мамонтовой, д. 5