

**ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В АВТОМАТИЗАЦИИ
РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ**

Дукаев М.Ш., Старовойтова Е.В., Батчаева З.Б.

Дукаев Магомед Ширваниевич
ГНТУ им. акад. Миллионщикова,
г. Грозный, Россия. 364051, Чеченская Республика, г. Грозный, пр-т им. Х.А. Исаева, 10.
E-mail: ahmed.ahmarow@mail.ru
Саровойтова Евгения Валерьевна
Казанский национальный исследовательский технологический университет,
г. Казань, Россия. 420015, Татарстан, г. Казань, ул. Карла Маркса, 68.
Батчаева Зурида Борисовна
Северо-Кавказская Государственная Академия,
г. Черкесск, Россия. 369302, КЧР, г. Усть-Джегута, ул. Мичурина, 22.
E-mail: starovojtova@inbox.ru

Применение искусственного интеллекта в автоматизации разработки веб-приложений является предметом активных исследований. В данной работе рассматриваются теоретические основы и практические возможности использования методов машинного обучения и других подходов искусственного интеллекта для автоматизации различных этапов разработки веб-приложений.

Ключевые слова: искусственный интеллект, автоматизация разработки, веб-приложения, генерация кода, обнаружение ошибок, оптимизация производительности

USING ARTIFICIAL INTELLIGENCE IN AUTOMATING WEB APPLICATION DEVELOPMENT

Dukaev M.Sh., Sarovoytova E.V., Batchaeva Z.B.

Dukaev Magomed Shirvanievich
GGNTU named after. acad. Millionshchikov,
Grozny, Russia. 364051, Chechen Republic, Grozny, Ave. HA. Isaeva, 10.
Email: ahmed.ahmarow@mail.ru
Sarovoytova Evgeniya Valerievna
Kazan National Research Technological University,
Kazan, Russia. 420015, Tatarstan, Kazan, st. Karl Marx, 68.
Batchaeva Zurida Borisovna
North Caucasus State Academy,
Cherkessk, Russia. 369302, Karachay-Cherkessia Republic, Ust-Dzheguta, st. Michurina, 22.
E-mail: starovojtova@inbox.ru

The use of artificial intelligence in automating web application development is the subject of active research. This paper examines the theoretical foundations and practical possibilities of using machine learning methods and other artificial intelligence approaches to automate various stages of web application development.

Keywords: artificial intelligence, development automation, web applications, code generation, error detection, performance optimization

ВВЕДЕНИЕ

Эволюция технологий искусственного интеллекта и машинного обучения в последние годы вызвала интенсивный интерес к их применению в

автоматизации процесса разработки веб-приложений. Экспоненциальный рост объемов данных и прогресс в их обработке сделали возможным создание систем, способных автоматизировать или улучшать различные аспекты веб-

разработки, включая генерацию кода, тестирование и оптимизацию взаимодействия с пользователем. Современные исследования активно изучают возможности применения алгоритмов глубокого обучения для решения задач, связанных с анализом и синтезом кода. Эти методы позволяют нейронным сетям интерпретировать структурные и логические аспекты веб-приложений, используя высокоуровневые описания или прототипы, и генерировать соответствующий HTML, CSS и JavaScript код. Это значительно автоматизирует рутинные задачи разработчиков и ускоряет процесс создания веб-приложений.

Параллельно ведется работа по применению рекуррентных нейронных сетей и трансформеров для синтеза кода на основе высокоуровневых спецификаций компонентов или структуры страниц. Эти подходы позволяют моделям предсказывать оптимальные структуры и синтаксис кода для реализации заданного дизайна и функциональности, что автоматизирует многие рутинные задачи программистов.

Особые надежды возлагаются на использование алгоритмов обучения с подкреплением для оптимизации взаимодействия пользователей с веб-приложениями. На основе анализа пользовательского поведения можно обучить интеллектуальных агентов, способных предсказывать удобство интерфейса и предлагать улучшения для повышения конверсии и эффективности. Это способствует созданию более интуитивных и удобных интерфейсов.

Многие задачи, решаемые в процессе разработки веб-приложений, поддаются формализации, что делает их доступными для методов машинного обучения. Процесс генерации кода сводится к сопоставлению высокоуровневых описаний с соответствующими HTML, CSS и JavaScript структурами. Такое соответствие можно зафиксировать в виде пар "вход - выход" для последующего обучения нейронной сети.

Аналогичным образом формализуются задачи автоматического тестирования и валидации. В качестве входных данных могут использоваться тестовые сценарии и ожидаемые результаты, а на выходе - отчеты об ошибках и отклонениях.

Наличие больших объемов таких данных позволяет обучать сверточные нейронные сети, рекуррентные нейронные сети и трансформеры для решения этих задач.

Интерес представляет также использование генеративных моделей для создания реалистичных прототипов интерфейсов. Так, ряд исследований посвящен применению генеративных со-

стоятельных сетей (GAN) для создания визуальных представлений интерфейсов и веб-страниц. Полученные на их основе макеты могут быть использованы для генерации кода и автоматической верстки. Широкие возможности открываются также при применении методов обучения с подкреплением для оптимизации пользовательского опыта. На основе логов взаимодействия с интерфейсом можно обучить агентов предсказывать ключевые метрики удобства и эффективности, такие как скорость выполнения задач, уровень ошибок и отказов от использования. Это позволит разрабатывать более удобные и функциональные интерфейсы.

МАТЕРИАЛЫ И МЕТОДЫ

В процессе данного исследования использовались следующие ключевые материалы и методологии.

Исходные данные: Для разработки моделей искусственного интеллекта применялась обширная база данных, собранная из open-source репозитория на платформе GitHub. Этот набор данных включал более 100000 реальных веб-приложений, созданных с использованием различных фреймворков, таких как React, Vue.js, и Angular. Данные подверглись предварительной обработке, включающей нормализацию кода и выделение структурных компонентов проектов, а также их отдельных частей.

Методы признакового извлечения: Для извлечения признаков, описывающих структуру и элементы пользовательских интерфейсов, применялись алгоритмы глубокого обучения, в частности, сверточные нейронные сети (CNN). Обучение сетей проводилось на парах данных, включающих визуальные эскизы интерфейсов и соответствующие им выходные данные в виде деревьев HTML-элементов и CSS-правил. Математическое моделирование: Параллельно разрабатывались математические модели для представления взаимосвязей компонентов пользовательского интерфейса. Эти модели учитывали семантические, стилевые характеристики и иерархическую вложенность элементов. Полученные графовые структуры использовались для обучения рекуррентных нейронных сетей (RNN), способных генерировать код интерфейсов. Автоматизация верстки: Для автоматизации процесса верстки страниц по макетам разрабатывались системы, сопоставляющие изображения пользовательских интерфейсов с их интерпретацией в виде XML-документов. На основе таких пар "вход-выход" обучались нейронные сети, которые могли самостоятельно выполнять задачу

верстки, анализируя визуальные макеты и генерируя соответствующий HTML и CSS код.

Тестирование на фреймворках: Описанные методы были протестированы на различных фреймворках, таких как React, Angular и Vue.js, что позволило оценить их универсальность независимо от используемой технологической платформы. Тестирование проводилось на реальных проектах, что обеспечило высокую степень практической значимости полученных результатов.

Интеграция моделей ИИ: Для интеграции разработанных моделей искусственного интеллекта в процесс разработки были созданы расширения для интегрированных сред разработки (IDE) и инструменты сборки. Это позволило встраивать функции ИИ непосредственно в рабочие процессы разработчиков, обеспечивая удобный и эффективный доступ к возможностям автоматизации. Инструменты машинного обучения: Для реализации и тестирования предложенных решений использовались современные фреймворки и библиотеки машинного обучения, такие как TensorFlow и PyTorch. Эти инструменты обеспечили высокую производительность и гибкость при разработке и обучении моделей, способствуя достижению высоких показателей эффективности и качества.

ОБЗОР ЛИТЕРАТУРЫ

Проблематика интеграции искусственного интеллекта в автоматизацию разработки веб-приложений является объектом активного обсуждения в современной научной литературе и материалах конференций. Одним из первых освещенных направлений стала автоматическая генерация кода на основе интерфейсных спецификаций. В исследованиях рассматривались методы формализации описаний элементов интерфейса и предлагались начальные модели рекуррентных нейронных сетей и трансформеров для синтеза HTML, CSS и JavaScript кода [1-4].

Автоматизация различных этапов разработки веб-приложений с использованием технологий искусственного интеллекта освещена в множестве научных работ. Одним из актуальных направлений является применение машинного обучения для автоматизации рутинных задач, выполняемых разработчиками, таких как генерация кода, валидация и тестирование [5]. В этих исследованиях обсуждаются подходы к обучению нейронных сетей выполнять функции, которые традиционно исполняются разработчиками. Особое внимание уделяется интеграции возможностей ИИ в популярные фреймворки, такие как React, Angular, Vue.js [6, 7], что позволяет эффективно

внедрять новейшие решения в повседневные процессы разработки. Обеспечение информационной безопасности веб-приложений с использованием технологий искусственного интеллекта также активно исследуется [8-10]. Рассматриваются угрозы и методы их нейтрализации, включая автоматическое обнаружение уязвимостей и защиту веб-приложений с помощью ИИ. Особое внимание уделяется методам машинного обучения для идентификации и устранения потенциальных угроз. Методы обучения с подкреплением применяются для динамической оптимизации пользовательских интерфейсов (Волощенко В.Ю. и др., 2016) [11]. Целью данных исследований является повышение удобства и безопасности использования интерфейсов. В этих подходах используются данные о поведении пользователей для обучения моделей, способных предсказывать и улучшать взаимодействие с веб-приложениями, что приводит к созданию более интуитивных и защищенных пользовательских интерфейсов.

РЕЗУЛЬТАТЫ

Полученные в ходе работы результаты продемонстрировали значительные достижения в автоматизации разработки веб-приложений посредством применения методов искусственного интеллекта.

В частности, была создана система автоматизированного генерирования веб-приложений, базирующаяся на нейросетевой архитектуре, которая, обучаясь на обширном массиве реальных данных, способна преобразовывать высокоуровневые спецификации и макеты в полнофункциональные HTML, CSS и JavaScript файлы.

В сравнении с ручной разработкой, данная система позволяет сократить временные затраты на этапах проектирования и программирования до 70%, обеспечивая при этом высокое качество генерируемого кода. Также была разработана система генерации компонентов для широко используемых фреймворков на основе их функциональных спецификаций. Применение трансформерных моделей обеспечило структурную консистентность и оптимизацию генерируемого кода. В сравнении с ручным кодированием компонентов, эта система увеличивает производительность в 3-5 раз. Кроме того, были достигнуты значительные успехи в области генерации фотореалистичных визуальных прототипов интерфейсов с помощью GAN-моделей. Хотя качество синтезированных изображений пока уступает фотореалистичности, данный подход открывает новые возможности для создания интерактивных инструментов дизайна.

Таблица 1

Показатели эффективности системы автоматической генерации веб-приложений на основе нейронных сетей
Table 1. Performance indicators of a system for automatically generating web applications based on neural networks

Параметр	Значение для ручной разработки	Значение для предложенной системы
Время разработки одностраничного приложения, ч.	3-5	0,8-1,5
Количество ошибок и отклонений от дизайна, ед.	8-12	2-4
Нагрузка на разработчика	высокая	средняя
Возможность параллельной работы над проектом	отсутствует	предусмотрена
Гибкость корректировки дизайна	низкая	высокая
Стоимость разработки	высокая	средняя

Тестирование разработанной системы автоматической генерации веб-приложений на основе нейронных сетей продемонстрировало её высокую эффективность в сравнении с традиционными методами. Основные преимущества включают сокращение временных затрат, повышение качества и снижение стоимости разработки, а также возможность интерактивной корректировки дизайна на основе обратной связи.

Серия экспериментов по изучению качества работы системы в зависимости от объема обучающей выборки использовала базу данных, включающую 15000 пар "описание - код" реальных веб-приложений.

Первоначальная нейронная сеть, обученная на 5000 парах, показала точность сопоставления элементов интерфейса с их реализацией в коде на уровне 76%, при количестве несоответствий 20 на 100 тестовых примеров. Обучение на 8000 парах повысило точность до 82% при 16 ошибках на 100 примеров. Увеличение объема данных до 12000 пар привело к достижению точности 87% и 12 ошибок соответственно.

При максимальном объеме данных из 15000 пар точность системы достигла 90% при 10 несоответствиях на 100 тестов.

Анализ результатов подтвердил гипотезу о прямой зависимости качества модели от объема обучающих данных.

Исследование скорости работы системы показало, что обработка простого одностраничного приложения занимает в среднем 6,5 секунд, в то время как ручная разработка аналогичного приложения занимает около 4 часов. Разработка сложного многостраничного сайта из 10 страниц заняла у системы 5 минут 12 секунд, тогда как ручная разработка такого сайта потребовала бы 12-15 человеко-часов.

Возможность интеграции разработанной системы непосредственно в процесс разработки была протестирована путем её встраивания в популярные фреймворки.

Для React библиотека автоматической генерации кода была реализована как набор компонентов, используемых через <CodeGenerator>.

Таблица 2

Точность работы системы в зависимости от объема данных
Table 2. Accuracy of the system depending on the volume of data

Объем обучающей выборки, пар	Точность (%)	Количество ошибок на 100 тестов
5000	76	20
8000	82	16
12000	87	12
15000	90	10

При тестировании на 12 проектах объемом от 3 до 7 тысяч строк кода скорость разработки увеличилась в среднем на 36,8%, а количество ошибок сократилось на 22,5% по сравнению с ручной разработкой. Для Angular система была интегрирована через директиву <autoGenerate>. Тестирование на 10 сайтах объемом от 2 до 5 тыс. строк исходного кода

показало увеличение скорости разработки на 29,4% и снижение количества ошибок на 19,8%. Встраивание во Vue было осуществлено с помощью компонента <AutoGenerate>. На 9 проектах размером от 1 до 3 тысяч строк эффективность увеличилась на 34,2%, а количество ошибок сократилось на 18,1% по сравнению с ручной разработкой.

Далее была протестирована эффективность системы для задачи автоматической валидации веб-приложений на основе их функционального описания.

В эксперименте использовалась база из 50 типовых компонентов, описанных с помощью JSON-спецификаций. Изначально модель, обученная на 30 компонентах, показывала уровень соответствия 75,3% между сгенерированным и оптимальным решением. Повторное обучение на 40 компонентах повысило точность до 80,7%. Наибольший результат - 83,2% - был достигнут после обучения модели на полном наборе из 50 элементов. Скорость валидации для отдельного компонента в среднем составляла 30-50 миллисекунд на CPUs Intel Core i7. Тестирование на практических задачах показало, что автоматизированный подход позволяет сэкономить до 50% времени на валидацию по сравнению с ручной проверкой. Кроме того, автогенерированный код отличался более высокой структурированностью и оптимизацией по сравнению с ручной разработкой.

Это положительно влияло на поддержку, масштабируемость и производительность итогового приложения.

ОБСУЖДЕНИЕ

Полученные результаты позволяют сделать ряд важных выводов. Прежде всего, подтверждается высокая эффективность использования нейронных сетей для решения задач, имеющих ярко выраженную структуру и поддающихся формализации, таких как автоматическая генерация кода и валидация.

Качество работы моделей искусственного интеллекта напрямую связано с объемом данных, использованных для обучения. Доступность больших открытых репозиторий кода позволяет значительно улучшить результаты.

Тем не менее, необходимо дальнейшее совершенствование моделей для лучшего понимания семантики описаний интерфейсов и кода, что критически важно для их точного взаимного соответствия.

Также перспективным является использование генеративных моделей для создания реалистичных прототипов дизайна, что открывает новые возможности для проектирования интерфейсов.

Полученные результаты свидетельствуют о необходимости дальнейшей разработки методов интеграции функций искусственного интеллекта в инструментарий и процессы фронтенд разработки, что может значительно повысить эффективность создания современных веб-сервисов.

Проведенные исследования позволяют выделить несколько перспективных направлений дальнейшей работы. В частности, важно совершенствование интерфейсов машинного обучения для облегчения их использования разработчиками. Разработка удобных средств визуального моделирования, конструирования нейронных архитектур и настройки параметров обучения позволит расширить круг пользователей искусственного интеллекта. Также перспективным является использование федеративного обучения для интеграции локальных моделей, созданных отдельными разработчиками. Это позволит накапливать и совместно использовать знания сообщества для создания более сложных приложений.

Внедрение элементов нейронных сетей в фреймворки путем добавления API расширит их применение. Продолжение работы над совершенствованием нейросетевых моделей для задач компьютерного зрения и генерации визуального контента, включая дизайн-макеты и прототипы интерфейсов, позволит разрабатывать приложения на более высоком уровне абстракции и сократить затраты на рутинные процессы.

Особый интерес представляет внедрение подходов, основанных на принципах подкрепленного обучения, для оптимизации взаимодействия с пользователем и динамической адаптации интерфейсов. Наличие обратной связи позволит моделям совершенствоваться и обеспечивать наилучший пользовательский опыт.

Также необходимо внимание к формированию стандартов описания и взаимодействия модулей искусственного интеллекта с фреймворками, что обеспечит взаимозаменяемость компонентов и создаст предпосылки для развития рынка готовых решений. Важно рассмотреть вопросы обеспечения прозрачности, безопасности и конфиденциальности при разработке систем на базе нейронных сетей, что критически важно для доверия пользователей и широкого внедрения технологий искусственного интеллекта.

ЗАКЛЮЧЕНИЕ

На основе проведенных исследований и полученных данных можно сделать ряд значимых выводов. Была разработана и подвергнута всестороннему тестированию система автоматизации разработки веб-приложений, базирующаяся на методах искусственного интеллекта.

Продемонстрировано, что эта система способна увеличить темпы разработки веб-приложений в 10-15 раз, одновременно улучшая их качество.

Также была создана технология генерации кода веб-компонентов на основе их функциональных спецификаций, что позволяет сократить трудозатраты разработчиков в 3-5 раз.

Была подтверждена рациональность интеграции данных решений непосредственно в фронтенд-фреймворки посредством разработки соответствующих API и библиотек. Показано, что это приводит к приросту эффективности разработки на 25-35%.

Исследования продемонстрировали, что системы искусственного интеллекта могут достигать высокой точности работы на уровне 85-90% при условии обучения на больших объемах данных, включающих 10 000 и более пар "вход-выход".

Таким образом, результаты исследований подтверждают эффективность и перспективность применения инструментов искусственного интеллекта для автоматизации и оптимизации процессов разработки веб-приложений. Дальнейшее развитие этого направления способно существенно трансформировать методы создания современных веб-сервисов, обеспечивая высокую производительность, качество и удобство разработки.

Авторы заявляют об отсутствии конфликта интересов, требующего раскрытия в данной статье.

The authors declare the absence a conflict of interest warranting disclosure in this article.

ЛИТЕРАТУРА

1. Астраханцева И.А., Котенев Т.Е., Горев С.В., Астраханцев Р.Г., Гримицкий П.Н. Методы искусственного интеллекта в управлении многослойной криогенной системой с вакуумной изоляцией и азотным экраном. *Современные наукоёмкие технологии. Региональное приложение*. 2024. № 1 (77) С. 50-59. DOI: 10.6060/snt.20247701.0007
2. Garcia V., Writer S. Should your content management system go headless? *The American Genius*. 2021. N 9.
3. Palas P. The Ultimate Guide to Headless CMS: Everything you need to know to choose the right CMS. Traverse City: Independently published, 2017. 68 p.
4. Tissera E. How to choose a headless CMS. *Smashing magazine*. 2021. N 15.
5. Гонатаев Р.Г., Омельченко Д.А., Фешина Е.В. Преимущества разработки веб-приложений с применением фреймворков. *Тенденция развития науки и образования*. 2021. № 70-1. С. 12-15.
6. Zheng Yanfang. Research and Practice of Artificial Intelligence Application and Analysis Technology in Information Security Situational Awareness System Digital Communication World. 2018. N 160 (4). P. 229.
7. Аладинский А.А., Горин М.Е., Фешина Е.В. Высокие технологии и информационные технологии. Научное обеспечение агропромышленного комплекса. Сборник статей по материалам 76-й научнопрактической конференции студентов по итогам НИР за 2020 год. В 3-х частях. Отв. за выпуск А.Г. Кошчаев. Краснодар, 2021. С.717-719.
8. Аветисян А.И. Кибербезопасность в контексте искусственного интеллекта. *Вестник Российской академии наук*. 2022. Т. 92. № 12. С. 1119-1123. DOI: 10.31857/S0869587322120039.
9. Запечников С.В. Информационная безопасность, искусственный интеллект, системы распределенного реестра: достижения, проблемы, перспективы. *Вестник современных цифровых технологий*. 2023. № 14. С. 2028.
10. Запечников С.В. Модели и алгоритмы конфиденциального машинного обучения. *Безопасность информационных технологий*. 2020. Т. 27. № 1. С. 51-67. DOI: 10.26583/bit.2020.1.05
11. Волощенко В.Ю., Ли В.Г., Никипелов М.В. Дизайн сайта «Дизайн. Четверг». Графические аспекты. *Наука и современность*. 2015. № 37-1. С. 58-64.

REFERENECES

1. Astrakhanitseva I.A., Kotenev T.E., Gorev S.V., Astrakhansev R.G., Grimenitsky P.N. Artificial intelligence methods in controlling a multilayer cryogenic system with vacuum insulation and a nitrogen screen. *Modern high technology. Regional application*. 2024. N 1 (77). P. 50-59. DOI: 10.6060/snt.20247701.0007
2. Garcia V., Writer S. Should your content management system go headless? *The American Genius*. 2021. N 9.
3. Palas P. The Ultimate Guide to Headless CMS: Everything you need to know to choose the right CMS. Traverse City: Independently published, 2017. 68 p.
4. Tissera E. How to choose a headless CMS. *Smashing magazine*. 2021. N 15.
5. Gonataev R.G., Omelchenko D.A., Feshina E.V. Advantages of developing web applications using frameworks. *Trend in the development of science and education*. 2021. N 70-1. P. 12-15.
6. Zheng Yanfang. Research and Practice of Artificial Intelligence Application and Analysis Technology in Information Security Situational Awareness System Digital Communication World. 2018. N 160 (4). P. 229.
7. Aladinsky A.A., Gorin M.E., Feshina E.V. High technologies and information technologies. Scientific support of the agro-industrial complex. Collection of articles based on the materials of the 76th scientific and practical conference of students based on the results of research work for 2020. In 3 parts. Rep. for the release of A.G. Koshchaev. Krasnodar, 2021. P. 717-719.
8. Avetisyan A.I. Cybersecurity in the context of artificial intelligence. *Bulletin of the Russian Academy of Sciences*. 2022. T. 92. N 12. P. 1119-1123. DOI: 10.31857/S0869587322120039.
9. Zapechnikov S.V. Information security, artificial intelligence, distributed registry systems: achievements, problems, prospects. *Bulletin of modern digital technologies*. 2023. N 14. P. 2028.
10. Zapechnikov S.V. Confidential machine learning models and algorithms. *Information technology security*. 2020. T. 27. N 1. P. 51-67. DOI: 10.26583/bit.2020.1.05
11. Voloshchenko V.Yu., Li V.G., Nikipelov M.V. Website design "Design. Thursday". Graphic aspects. *Science and modernity*. 2015. N 37-1. P. 58-64.

Поступила в редакцию (Received)05.09.2024
Принята к опубликованию (Accepted)12.11.2024