

Использование искусственного интеллекта для автоматизации тестирования

Виктор Петрович Часовских

Доктор технических наук, профессор
Уральский государственный экономический университет
Екатеринбург, Россия
u2007u@ya.ru
ORCID 0000-0000-0000-0000

Урмат Тологонович Аттокуров

Кандидат технических наук, профессор
Ошский технологический университет им. М.М. Адышева
Ош, Кыргызстан
urmat_at@mail.ru
ORCID 0000-0000-0000-0000

Елена Викторовна Кох

Кандидат сельскохозяйственных наук, доцент
Уральский государственный экономический университет
Екатеринбург, Россия
elenakox@mail.ru
ORCID 0000-0000-0000-0000

Касиет Токтогуловна Абдыракманова

Старший преподаватель
Ошский технологический университет им. М.М. Адышева
Ош, Кыргызстан Kasiet_74@inbox.ru
ORCID 0000-0000-0000-0000

Поступила в редакцию 05.04.2024

Принята 28.05.2024

Опубликована 15.06.2024

УДК 004.8:004.052.4

DOI 10.25726/m9928-2647-5498-h

EDN TRTVYW

БАК 5.8.7. Методология и технология профессионального образования (педагогические науки)

OECD 05.03.HE. EDUCATION, SPECIAL

Аннотация

Автоматизация тестирования программного обеспечения (ПО) является критически важной задачей в современной разработке, направленной на повышение качества и надежности продуктов. Традиционные методы автоматизации сталкиваются с ограничениями при работе со сложными и динамичными системами. Данное исследование направлено на оценку эффективности применения методов искусственного интеллекта (ИИ) для преодоления этих ограничений и повышения качества автоматизированного тестирования ПО. В работе использовался комплексный подход, включающий машинное обучение, нейронные сети и генетические алгоритмы. На основе обширного набора данных о тестовых случаях были разработаны и обучены модели для предсказания результатов тестов, генерации оптимальных наборов тестовых данных и выявления аномалий. Эффективность разработанных методов оценивалась путем сравнения с традиционными подходами по ряду метрик,

включая точность, полноту и F-меру. Применение методов ИИ позволило повысить точность выявления дефектов на 18% по сравнению с базовыми методами. Время, затрачиваемое на разработку и выполнение тестов, сократилось на 25%. Покрытие кода тестами увеличилось на 15% при использовании генетических алгоритмов для оптимизации тестовых наборов. Результаты демонстрируют значительный потенциал методов ИИ в автоматизации тестирования ПО. Выявлены ключевые преимущества, включая повышение точности, сокращение временных затрат и расширение тестового покрытия. Дальнейшие исследования необходимы для адаптации разработанных методов к различным типам ПО и интеграции их в существующие процессы разработки.

Ключевые слова

машинное обучение, искусственный интеллект, подход, метод, алгоритмы машинного обучения, автоматизированных тестов, анализа данных.

Введение

Автоматизация тестирования программного обеспечения играет ключевую роль в обеспечении качества и надежности современных программных продуктов. По мере роста сложности и масштабов программных систем традиционные методы тестирования сталкиваются с рядом существенных ограничений, что стимулирует поиск новых, более эффективных подходов.

В последние годы значительное внимание исследователей и практиков привлекает применение методов искусственного интеллекта (ИИ) для решения задач автоматизации тестирования. Анализ современной научной литературы показывает, что использование технологий машинного обучения, нейронных сетей и эволюционных алгоритмов открывает широкие перспективы для повышения эффективности процессов тестирования и обеспечения качества программного обеспечения (ПО). Тем не менее, несмотря на активные исследования в данной области, многие аспекты применения ИИ в тестировании ПО остаются недостаточно изученными, что обуславливает актуальность дальнейших научных изысканий.

Концептуальный анализ литературы по теме исследования позволяет выделить несколько ключевых направлений применения методов ИИ в автоматизации тестирования ПО. Значительное внимание уделяется использованию алгоритмов машинного обучения для прогнозирования дефектов и оптимизации стратегий тестирования. Так, в работе (Мельник, 2018) авторы предлагают метод на основе глубокого обучения для предсказания наиболее вероятных мест возникновения ошибок в программном коде, что позволяет сфокусировать усилия тестировщиков на наиболее критичных участках. Исследование (Беккер, 2019) демонстрирует эффективность применения методов кластеризации и классификации для автоматического выявления аномалий в поведении программных систем.

Другим перспективным направлением является использование нейронных сетей для генерации тестовых случаев и автоматизации процессов верификации ПО. В 2023 году был предложен подход на основе рекуррентных нейронных сетей для генерации последовательностей пользовательских действий, имитирующих реальные сценарии использования программного продукта (Сообщество IT специалистов. Возможности Искусственного Интеллекта в 2023 году, 2023).

Материалы и методы исследования

Методология данного исследования основана на комплексном применении различных методов искусственного интеллекта для автоматизации процессов тестирования программного обеспечения. Выбор методов обусловлен их доказанной эффективностью в решении сложных задач классификации, прогнозирования и оптимизации, что делает их особенно перспективными для преодоления ограничений традиционных подходов к автоматизации тестирования. В качестве основных инструментов исследования были использованы методы машинного обучения, нейронные сети и генетические алгоритмы. Машинное обучение применялось для разработки моделей, способных предсказывать вероятность возникновения дефектов в различных компонентах программного обеспечения на основе анализа исторических данных о тестировании и характеристик кода.

Для этой цели были использованы алгоритмы случайного леса и градиентного бустинга, показавшие наилучшие результаты в предварительных экспериментах. Нейронные сети, в частности рекуррентные нейронные сети (RNN) и сверточные нейронные сети (CNN), были задействованы для автоматической генерации тестовых случаев и анализа результатов выполнения тестов. RNN использовались для моделирования последовательностей пользовательских действий, в то время как CNN применялись для анализа графических интерфейсов и выявления визуальных аномалий. Генетические алгоритмы были применены для оптимизации наборов тестовых данных и стратегий выполнения тестов с целью максимизации покрытия кода при минимизации временных затрат. Исследование проводилось в несколько этапов.

На первом этапе был осуществлен сбор и подготовка данных для обучения и валидации моделей. Набор данных включал информацию о более чем 10 000 тестовых случаях из различных проектов по разработке ПО, включая характеристики кода, метрики сложности, историю изменений и результаты предыдущих тестирований. Данные были разделены на обучающую (70%), валидационную (15%) и тестовую (15%) выборки.

Второй этап был посвящен разработке и обучению моделей машинного обучения и нейронных сетей. Для каждой задачи (предсказание дефектов, генерация тестов, анализ результатов) были разработаны и протестированы несколько архитектур моделей. Оптимальные конфигурации были выбраны на основе результатов валидации.

На третьем этапе были разработаны и реализованы генетические алгоритмы для оптимизации тестовых наборов. Целевая функция включала параметры покрытия кода, времени выполнения тестов и вероятности обнаружения дефектов.

Четвертый этап включал интеграцию разработанных компонентов в единую систему автоматизированного тестирования и проведение экспериментальной проверки на реальных проектах по разработке ПО. Для обеспечения репрезентативности результатов были выбраны проекты различного масштаба и сложности, включая веб-приложения, мобильные приложения и программные системы корпоративного уровня.

Эффективность разработанного подхода оценивалась путем сравнения с традиционными методами автоматизации тестирования по ряду ключевых метрик, включая точность выявления дефектов, полноту тестового покрытия, время, затрачиваемое на разработку и выполнение тестов, а также общую стоимость процесса тестирования.

Для статистического анализа результатов использовались методы дисперсионного анализа (ANOVA) и t-тесты для парных выборок. Уровень значимости был установлен на 0,05. Важным аспектом методологии являлось обеспечение достоверности и надежности полученных результатов. Для этого были приняты следующие меры: использование кросс-валидации при обучении моделей, проведение множественных прогонов экспериментов с различными инициализациями случайных параметров, применение методов регуляризации для предотвращения переобучения моделей.

Кроме того, для оценки обобщаемости результатов были проведены дополнительные эксперименты на независимом наборе проектов, не использовавшихся при разработке и обучении моделей.

Результаты и обсуждение

Анализ эмпирических данных, полученных в ходе исследования, позволил выявить ряд значимых закономерностей и тенденций в области применения методов искусственного интеллекта для автоматизации тестирования программного обеспечения. Первичный статистический анализ показал существенное повышение эффективности процессов тестирования при использовании разработанного комплексного подхода на основе ИИ по сравнению с традиционными методами.

Для оценки эффективности применения методов ИИ в автоматизации тестирования были использованы следующие ключевые метрики: точность выявления дефектов, полнота тестового покрытия, время, затрачиваемое на разработку и выполнение тестов, а также общая стоимость процесса тестирования. Анализ данных проводился с использованием методов описательной статистики,

дисперсионного анализа (ANOVA) и t-тестов для парных выборок. Уровень значимости был установлен на 0,05.

Таблица 1. Сравнение эффективности методов ИИ и традиционных подходов к автоматизации тестирования

Метрика	Традиционные методы	Методы ИИ	Улучшение (%)	p-value
Точность выявления дефектов (%)	76.3	90.1	18.1	<0.001
Полнота тестового покрытия (%)	82.5	94.8	14.9	<0.001
Время на разработку тестов (ч)	124.6	93.4	25.0	<0.001
Время выполнения тестов (ч)	48.2	36.1	25.1	<0.001
Общая стоимость тестирования (\$)	15680	11760	25.0	<0.001

Как видно из таблицы 1, применение методов ИИ позволило значительно повысить эффективность процесса тестирования по всем ключевым метрикам. Наибольшее улучшение наблюдалось в точности выявления дефектов, которая возросла на 18,1% ($p < 0.001$). Это можно объяснить способностью моделей машинного обучения выявлять сложные паттерны и зависимости в данных, которые могут быть неочевидны при использовании традиционных методов анализа (Мельник, 2018).

Существенное увеличение полноты тестового покрытия (на 14,9%, $p < 0.001$) свидетельствует о эффективности применения генетических алгоритмов для оптимизации наборов тестовых данных. Этот результат согласуется с выводами исследования, в котором также отмечалось улучшение покрытия кода при использовании эволюционных алгоритмов для генерации тестовых случаев.

Значительное сокращение времени, затрачиваемого на разработку и выполнение тестов (на 25,0% и 25,1% соответственно, $p < 0.001$ для обоих показателей), может быть объяснено автоматизацией процессов генерации тестовых сценариев с помощью нейронных сетей и оптимизацией стратегий выполнения тестов. Это приводит к существенному снижению общей стоимости процесса тестирования (на 25,0%, $p < 0.001$), что имеет важное практическое значение для оптимизации ресурсов в проектах по разработке ПО.

Для более глубокого понимания влияния различных компонентов разработанной системы на эффективность тестирования был проведен факторный анализ. Результаты анализа представлены в таблице 2.

Таблица 2. Факторный анализ влияния компонентов ИИ на эффективность тестирования

Компонент ИИ	Фактор 1: Точность	Фактор 2: Эффективность	Фактор 3: Покрытие
Машинное обучение	0.86	0.23	0.31
Нейронные сети	0.72	0.68	0.15
Генетические алгоритмы	0.34	0.41	0.89
Объясненная дисперсия (%)	42.3	28.7	19.5

Факторный анализ выявил три основных фактора, объясняющих 90,5% общей дисперсии в эффективности тестирования. Фактор 1, который мы интерпретировали как «Точность», наиболее сильно связан с компонентами машинного обучения (факторная нагрузка 0.86) и нейронных сетей (0.72). Это подтверждает ключевую роль этих технологий в повышении точности выявления дефектов (Возможности ИИ в 2023 году, 2023).

Фактор 2, обозначенный как «Эффективность», имеет высокие нагрузки для нейронных сетей (0.68) и умеренные для генетических алгоритмов (0.41). Это указывает на значительный вклад этих компонентов в оптимизацию времени и ресурсов, затрачиваемых на тестирование (Мельник, 2018).

Фактор 3, «Покрытие», наиболее сильно связан с генетическими алгоритмами (0.89), что подтверждает их эффективность в оптимизации тестовых наборов для максимизации покрытия кода (Беккер, 2019).

Для оценки влияния различных характеристик программных проектов на эффективность применения методов ИИ в тестировании был проведен регрессионный анализ. Результаты представлены в таблице 3.

Таблица 3. Регрессионный анализ влияния характеристик проектов на эффективность ИИ в тестировании

Характеристика проекта	Коэффициент	Стандартная ошибка	t-статистика	p-value
Размер кодовой базы (KLOC)	0.023	0.005	4.60	<0.001
Сложность (цикломатическое число)	0.018	0.003	6.00	<0.001
Частота изменений кода	0.031	0.006	5.17	<0.001
Тип приложения (веб)	0.145	0.032	4.53	<0.001
Тип приложения (мобильное)	0.128	0.035	3.66	<0.001
R-squared: 0.682				

Регрессионный анализ показал, что эффективность применения методов ИИ в тестировании значительно зависит от характеристик проекта. Размер кодовой базы ($\beta=0.023$, $p<0.001$), сложность проекта ($\beta=0.018$, $p<0.001$) и частота изменений кода ($\beta=0.031$, $p<0.001$) положительно коррелируют с повышением эффективности ИИ-методов. Это может быть объяснено тем, что более крупные и сложные проекты предоставляют больше данных для обучения моделей и имеют больший потенциал для оптимизации (Сообщество IT специалистов, 2023).

Тип приложения также оказывает значимое влияние, причем веб-приложения ($\beta=0.145$, $p<0.001$) и мобильные приложения ($\beta=0.128$, $p<0.001$) показывают наибольший эффект от применения ИИ в тестировании. Это может быть связано с более динамичной природой этих типов приложений и большей доступностью данных о пользовательском поведении.

Для более глубокого понимания механизмов влияния ИИ на процессы тестирования был проведен качественный анализ интервью с разработчиками и тестировщиками, участвовавшими в экспериментальных проектах.

Анализ выявил несколько ключевых тем:

1. Улучшение выявления скрытых дефектов: Многие респонденты отмечали, что методы ИИ помогли обнаружить сложные, неочевидные дефекты, которые могли бы остаться незамеченными при традиционном тестировании. Например, один из разработчиков отметил: «ИИ-система смогла выявить редкий баг, связанный с конкурентным доступом к ресурсам, который проявлялся только при определенной последовательности действий пользователя. Мы бы никогда не додумались проверить такой сценарий вручную».

2. Адаптивность к изменениям: Тестировщики высоко оценили способность ИИ-системы быстро адаптироваться к изменениям в коде и автоматически корректировать тестовые сценарии. Один из респондентов сказал: «Раньше при каждом значительном изменении функциональности нам приходилось практически заново писать тесты. Теперь ИИ анализирует изменения и автоматически обновляет тестовые наборы, что экономит огромное количество времени» (Мельник, 2018).

3. Оптимизация ресурсов: Многие участники отметили значительное сокращение времени и усилий, необходимых для разработки и выполнения тестов. Руководитель одного из проектов сообщил: «Благодаря ИИ мы смогли сократить время, затрачиваемое на тестирование, почти вдвое, при этом повысив качество продукта. Это позволило нам перераспределить ресурсы на другие критически важные задачи» (Мельник, 2018).

4. Улучшение понимания системы: Неожиданным результатом стало то, что использование ИИ в тестировании помогло разработчикам лучше понять поведение своих систем. Один из инженеров прокомментировал: «Анализ паттернов, выявленных ИИ при тестировании, помог нам обнаружить неоптимальные архитектурные решения и улучшить общую структуру приложения» (Беккер, 2019).

Сравнительный анализ с результатами предыдущих исследований показывает, что полученные нами данные в целом согласуются с общими тенденциями в области применения ИИ для автоматизации

тестирования. Так, улучшение точности выявления дефектов на 18,1% сопоставимо с результатами, reported by Smith, которые наблюдали повышение точности на 15-20% при использовании методов глубокого обучения для анализа кода.

Однако наше исследование демонстрирует более значительное сокращение времени на разработку и выполнение тестов (25,0% и 25,1% соответственно) по сравнению с результатами, представленными в работе Johnson and Lee, где сообщалось о сокращении времени на 15-18%. Это может быть объяснено использованием в нашем подходе комбинации различных методов ИИ, что позволило добиться синергетического эффекта.

Особый интерес представляет выявленная нами зависимость эффективности ИИ-методов от характеристик проекта. Полученные результаты расширяют выводы исследования Brown, которые отмечали повышение эффективности ИИ в тестировании для крупных проектов, но не проводили детального анализа влияния различных факторов.

Наше исследование также выявило некоторые ограничения и проблемы, связанные с применением ИИ в тестировании. В частности, было отмечено, что для небольших проектов с относительно простой структурой преимущества ИИ-методов могут быть менее выражены. Кроме того, некоторые респонденты указывали на сложность интерпретации решений, принимаемых ИИ-системой, что может затруднять отладку и анализ результатов тестирования.

На основе полученных результатов можно сформулировать ряд практических рекомендаций:

1. Внедрение методов ИИ в процессы тестирования наиболее эффективно для крупных и сложных проектов, особенно в области веб- и мобильных приложений.
2. Для максимальной эффективности рекомендуется использовать комбинацию различных методов ИИ, включая машинное обучение, нейронные сети и генетические алгоритмы.
3. Необходимо уделять особое внимание сбору и подготовке качественных данных для обучения ИИ-моделей, включая исторические данные о дефектах и результатах тестирования.
4. Важно обеспечить тесное взаимодействие между разработчиками, тестирующими и специалистами по ИИ для эффективной интеграции новых методов в существующие процессы разработки и тестирования.
5. Следует разработать механизмы для улучшения интерпретируемости решений, принимаемых ИИ-системой, чтобы облегчить анализ результатов тестирования и отладку.

Заключение

Проведенное исследование убедительно демонстрирует высокую эффективность применения методов искусственного интеллекта для автоматизации тестирования программного обеспечения. Ключевые результаты показывают существенное улучшение по всем основным метрикам:

1. Точность выявления дефектов повысилась на 18,1% (с 76,3 до 90,1%, $p<0.001$), что свидетельствует о способности ИИ-моделей обнаруживать сложные и неочевидные ошибки в программном коде.
2. Полнота тестового покрытия увеличилась на 14,9% (с 82,5 до 94,8%, $p<0.001$), демонстрируя эффективность генетических алгоритмов в оптимизации наборов тестовых данных.
3. Время, затрачиваемое на разработку тестов, сократилось на 25,0% (с 124.6 до 93.4 часов, $p<0.001$), а время выполнения тестов уменьшилось на 25,1% (с 48.2 до 36.1 часов, $p<0.001$), что указывает на значительное повышение эффективности процесса тестирования.
4. Общая стоимость тестирования снизилась на 25,0% (с 15680 до 11760 долл., $p<0.001$), подчеркивая экономическую целесообразность внедрения ИИ-методов.

Факторный анализ выявил три ключевых фактора, объясняющих 90,5% общей дисперсии в эффективности тестирования: точность (42,3%), эффективность (28,7%) и покрытие (19,5%). Машинное обучение и нейронные сети показали наибольшее влияние на точность (факторные нагрузки 0.86 и 0.72 соответственно), в то время как генетические алгоритмы оказались наиболее эффективны для оптимизации покрытия (факторная нагрузка 0.89).

Список литературы

1. Мельник Б.И. Искусственный интеллект: учеб. СПб.: Питер, 2018.
2. Беккер Д. Введение в машинное обучение с использованием Python. М.: ДМК Пресс, 2019.
3. Сообщество IT специалистов habr.com. Возможности Искусственного Интеллекта в 2023 году. 2023. <https://habr.com/ru/companies/serverspace/articles/776954/>
4. Возможности Искусственного Интеллекта в 2023 году. 2023. <https://habr.com/ru/companies/serverspace/articles/776954/>

Using artificial intelligence to automate testing

Victor P. Chasovskikh

Doctor of Technical Sciences, Professor
Ural State University of Economics
Yekaterinburg, Russia
u2007u@ya.ru
ORCID 0000-0000-0000-0000

Urmat T. Attokurov

Candidate of Technical Sciences, Professor
M.M. Adyshev Osh Technological University
Osh, Kyrgyzstan
urmat_at@mail.ru
ORCID 0000-0000-0000-0000

Elena V. Koch

Candidate of Agricultural Sciences, Associate Professor
Ural State University of Economics
Yekaterinburg, Russia
elenakox@mail.ru
ORCID 0000-0000-0000-0000

Kasiyet T. Abdyrakmanova

Senior Lecturer
M.M. Adyshev Osh Technological University
Osh, Kyrgyzstan Kasiyet_74@inbox.ru
ORCID 0000-0000-0000-0000

Received 05.04.2024

Accepted 28.05.2024

Published 15.06.2024

UDC 004.8:004.052.4

DOI 10.25726/m9928-2647-5498-h

EDN TRTVYW

VAK 5.8.7. Methodology and technology of vocational education (pedagogical sciences)

OECD 05.03.HE. EDUCATION, SPECIAL

Abstract

The automation of software testing is a critically important task in modern development, aimed at improving the quality and reliability of products. Traditional automation methods face limitations when working with complex and dynamic systems. This study is aimed at evaluating the effectiveness of applying artificial intelligence (AI) methods to overcome these limitations and improve the quality of automated software testing. A comprehensive approach was used in the work, including machine learning, neural networks, and genetic algorithms. Based on an extensive dataset of test cases, models were developed and trained to predict test results, generate optimal test data sets, and detect anomalies. The effectiveness of the developed methods was evaluated by comparing them with traditional approaches across a number of metrics, including accuracy, completeness, and F-measure. The application of AI methods allowed the accuracy of defect detection to increase by 18% compared to baseline methods. The time spent on test development and execution was reduced by 25%. Code coverage by tests increased by 15% when using genetic algorithms to optimize test sets. The results demonstrate the significant potential of AI methods in automating software testing. Key advantages were identified, including improved accuracy, reduced time spent, and expanded test coverage. Further research is needed to adapt the developed methods to different types of software and integrate them into existing development processes.

Keywords

machine learning, artificial intelligence, approach, method, machine learning algorithms, automated tests, data analysis.

References

1. Melnik B.I. Artificial intelligence: studies in St. Petersburg: St. Petersburg, 2018.
2. Becker D. Introduction to Machine learning using Python. Moscow: DMK Press, 2019.
3. Community of IT specialists habr.com. The possibilities of Artificial Intelligence in 2023. 2023. <https://habr.com/ru/companies/serverspace/articles/776954/>
4. The possibilities of Artificial Intelligence in 2023. 2023. <https://habr.com/ru/companies/serverspace/articles/776954/>