

УДК 004.415.538

## ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ТЕСТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Киселёва М.П., Медведев О.С., Коркин Л.Р.

Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь

Научный руководитель: Меженная М.М. – канд. техн. наук, доцент, доцент кафедры ИПиЭ

**Аннотация.** Рассмотрены основные направления применения использования искусственного интеллекта в тестировании программного обеспечения. Также представлен краткий обзор уже существующего инструментария в данной сфере и преимущества, которые предоставляет данное направление тестирования.

**Ключевые слова:** тестирование, искусственный интеллект, машинное обучение

**Введение.** В процессе современной разработки программного обеспечения качество и скорость тестирования играет значительную роль. От него зависит работоспособность готового продукта, его безопасность, надежность, а также временные и денежные расходы на разработку. Разработка процессов, методов и методологий тестирования и управления программным обеспечением имеет решающее значение для минимизации или даже исключения ошибок в коде и, таким образом, для предотвращения негативных последствий, возникающих в результате использования программного обеспечения, подверженного ошибкам. Сегодня, по мере развития технологий, все больше и больше компаний внедряют методы *agile* и *DevOps*. Но с внедрением этих методологий также возрастает потребность в надежных инструментах, поддерживающих непрерывное тестирование и непрерывный выпуск. Для удовлетворения потребностей современной сферы разработки программного обеспечения появляются инструменты автоматизации тестирования на основе искусственного интеллекта.

**Основная часть.** Тестирование на основе искусственного интеллекта – это метод тестирования программного обеспечения, в котором алгоритмы искусственного интеллекта и машинного обучения используются для эффективного тестирования программного продукта.

В современном мире искусственный интеллект и машинное обучение при тестировании программного обеспечения обеспечивают лучшую и более эффективную автоматизацию, освобождая команды от необходимости повторять и совершенствовать тестирование. Многие методы тестирования программного обеспечения в настоящее время основаны на искусственном интеллекте и алгоритмах глубокого обучения.

Существует 4 подхода к Искусственному интеллекту при тестировании программного обеспечения.

Первым подходом является дифференциальное тестирование. Данный тип тестирования используется классификации различий и сравнения версий приложений для каждой сборки. Инструменты, поддерживающие данный подход, используют алгоритмы искусственного интеллекта и машинного обучения для выявления проблем, связанных с кодом, уязвимостей в системе безопасности, регрессий и так далее. Это достигается за счет сканирования кода, автоматизации модульного тестирования и так далее. Примером такого инструмента является *Launchable*. В основе данного инструмента лежит алгоритм машинного обучения, который предсказывает вероятность сбоя для каждого теста на основе прошлых запусков и всякий раз, когда исходный код изменяется в процессе тестирования. Он позволяет обеспечить первостепенный запуск тестов, с наибольшей вероятностью неудачи. Это позволяет значительно сократить время выполнения тестов.

Второй подход – визуальное тестирование. Визуальное тестирование – это метод тестирования программного обеспечения, при котором внешний вид приложения проверяется с помощью обучения на основе изображений и сравнения экрана. Среди современных

инструментов визуального тестирования можно выделить *Applitools*. *Applitools* – это популярный инструмент, поддерживающий тестирование визуальных элементов. Этот инструмент на базе искусственного интеллекта обнаруживает сбои и ошибки с точки зрения пользовательского интерфейса. Основные функции включают *Applitools Eyes*, который помогает увеличить охват тестированием и сократить расходы на техническое обслуживание. *Applitools Eyes* представляет из себя облачный сервис, позволяющий проводить визуальные валидации путем сравнения изображений. Он выполняет такие визуальные валидации, как проверка GUI, сравнение с эталонным изображением *Web*, *Mobile* и *Native* приложений. *Applitools* интегрируется со всеми современными тестовыми фреймворками и работает со многими существующими инструментами тестирования, такими как *Selenium*, *Appium* и *Cypress*.

Третий подход – декларативное тестирование. Он направлен на определение цели теста на естественном или специфичном для конкретной предметной области языке. Система решает, как выполнить тест. Декларативные инструменты направлены на повышение производительности и стабильности автоматизации тестирования. Эти инструменты обладают значительными возможностями, связанными с автоматизацией роботизированных процессов (*RPA*), обработкой естественного языка (*NLP*), автоматизацией тестирования на основе моделей (*MBTA*) и автономными методами тестирования (*AT*).

Основная цель этих методов – устранить утомительные, подверженные ошибкам, повторяющиеся задачи с помощью интеллектуальной автоматизации. Примером инструмента, который попадает под эту категорию, является *Tricentis*.

*Tricentis* – это модельно-ориентированный инструмент автоматизации тестирования, который предоставляет довольно широкий набор функций для непрерывного тестирования, включая тестирование с последующим выведением данных, их анализом и интеграцией для поддержки гибких методологий программирования и *DevOps*-методологий.

Четвертым подходом является автоматизация самовосстановления. Самовосстановление – это способность системы автоматически вносить необходимые изменения или модификации. Большинство инструментов на базе искусственного интеллекта обладают способностью как к самотестированию, так и к самовосстановлению, и они считаются ‘умными’. Главной идеей данного подхода является то, что выбор элемента в тестах автоматически корректируется при изменении пользовательского интерфейса.

В качестве примера инструмента данного подхода можно привести *Mabl* и *Testim*.

*Mabl* – это ведущая интеллектуальная платформа автоматизации тестирования, созданная для *CI / CD*. Этот инструмент использует машинное обучение и адаптивные возможности для экономии начнет функционировать в соответствии с вашим предпочтительным направлением действий.

*Testim* – инструмент, который использует алгоритмы искусственного интеллекта и *ML* для полной автоматизации тестирования. Искусственный интеллект используется для ускорения разработки, выполнения и обслуживания тестов. *Testim* включает в себя самообслуживание автоматизированных тестов, основанных на *ML*. Это приводит к быстрому созданию стабильных автоматизированных тестов.

Первым главным достоинством тестирования программного продукта с использованием ИИ является экономия времени. Требования к программным продуктам усложняются и вместе с этим увеличивается время, затрачиваемое тестировщиками на написание тестов.

Благодаря ИИ это время можно сократить вдвое, при этом не качество тестов не будет хуже. Следующим плюсом является уменьшение ошибок. Тесты, написанные машиной, с меньшей вероятностью содержат ошибки, чем тесты, написанные человеком. Также, с помощью тестов ИИ можно обнаружить ошибки на ранних этапах разработки, что уменьшает количество дефектов в времени и сокращения времени на написание сценариев кода. Все, что вам нужно сделать, это интегрировать ваш тест с программой, используя язык *Mabl*, после чего инструмент продукте, не допуская того, чтобы они дошли до конечного пользователя.

Тестирование с применением ИИ обладает возможностями распознавания образов и изображений, которые вместе помогают обнаруживать визуальные ошибки, выполняя визуальное тестирование приложений. Так можно убедиться, что все элементы привлекательны, должным образом функционирующие, а динамические элементы корректно работают независимо от их размера и формы. Более того, с помощью ИИ можно расширить область испытаний и получить полный анализ тестируемого компонента.

Тестирование с использованием ИИ обладает недостатками. Первый недостаток - это цена такого тестирования. Разработка инструмента на базе ИИ занимает большое количество времени и ресурсов. Данный подход имеет место быть, то в долгосрочной перспективе, так можно получить от него выгоду. Следующий минус связан со стереотипом, что машины заменят человеческий труд. Нужно понимать, что построение машин без человека невозможно, поэтому за это переживать не стоит: искусственный интеллект будет выполнять за человека его повседневные задачи, а не работать за него полностью.

**Заключение.** Таким образом в результате рассмотрения современных тенденций развития искусственного интеллекта в сфере тестирования, а также инструментов, которые существуют на данный момент на рынке можно отметить востребованность данного направления развития тестирования программного обеспечения. Существует ряд подходов тестирования с использованием искусственного интеллекта, каждый из которых имеет свою сферу применения и продукты, уже представленные на рынке. Инструменты автоматизации тестирования на основе искусственного интеллекта позволяют соответствовать требованиям качества и скорости, а также поддерживать непрерывное тестирование и непрерывный выпуск, которые требуются в условиях современной разработки программного обеспечения.

#### **Список литературы**

1. *AI in Software Testing – Benefits, Approaches, Tools to Look in 2022 [Electronic resource] – 2021. – Mode of access: <https://www.testingxperts.com/blog/AI-in-Software-Testing> – Date of access: 01.04.2022.*
2. *How Can Teams Apply AI and ML in Testing? Use Cases, Tools, and Benefits [Electronic resource] – 2021. – Mode of access: <https://utor.com/topic/ai-and-ml-in-software-testing> – Date of access: 01.04.2022.*
3. *AI in test automation: here’s how it works – Benefits, Approaches, Tools to Look in 2022 [Electronic resource] – 2021. – Mode of access: <https://theqalead.com/topics/ai-test-automation> – Date of access: 01.04.2022.*

UDC 004.415.538

## **THE APPLICATION OF ARTIFICIAL INTELLIGENCE IN SOFTWARE TESTING**

*Kiseliova M.P., Medvedev O.S., Korkin L.R.*

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

*Mezhennaya M.M. – PhD, assistant professor, associate professor of the department of EPE*

**Annotation.** The main directions of artificial intelligence application in software testing are addressed. A brief overview is presented regarding currently existing tools implemented in this area and the advantages provided by this type of testing.

**Keywords:** testing, Artificial Intelligence, Machine Learning