Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.8

Желтко
Юлия Юрьевна

ПРОГРАММНЫЙ КОМПЛЕКС ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА СТАТИСТИЧЕСКИХ ДАННЫХ

АВТОРЕФЕРАТ
на соискание степени магистра

по специальности 7-06-0611-05 Компьютерная инженерия. Хранение и обработка данных

Научный руководитель
Одинец Дмитрий Николаевич
кандидат технических наук,
доцент кафедры ЭВМ, БГУИР

Минск 2025

ВВЕДЕНИЕ

Современные системы искусственного интеллекта (ИИ) и машинного обучения (МО) активно используются в программировании, что способствует автоматизации процесса разработки программного обеспечения. В последние годы значительное распространение получили нейросетевые модели генерации кода, такие как OpenAI Codex, GitHub Copilot и DeepCode. Эти системы способны интерпретировать текстовые описания задач и автоматически генерировать программный код, что существенно ускоряет процесс разработки. Однако наряду с преимуществами использование таких технологий порождает ряд проблем, связанных с качеством, безопасностью, интеллектуальной собственностью и идентификацией источника кода.

Одним из наиболее актуальных вызовов является задача определения авторства программного кода, то есть разграничения кода, написанного человеком, и кода, сгенерированного нейросетевой моделью. Данная проблема становится всё более значимой в контексте обеспечения безопасности программного обеспечения, предотвращения уязвимостей и защиты интеллектуальной собственности. Генеративные модели обучаются на больших массивах открытых данных, что может приводить к распространению устаревших или уязвимых решений, а также к копированию существующего кода без учёта авторских прав.

В образовательной среде также возникает проблема автоматической генерации кода, когда студенты используют нейросетевые модели для выполнения учебных заданий, что затрудняет объективную оценку их знаний и навыков. В связи с этим необходимы методы анализа программного кода, позволяющие автоматически определять его происхождение.

В рамках данной работы основное внимание уделяется сравнительному анализу исходного кода, написанного человеком, и кода, сгенерированного нейросетевыми моделями. Цель работы состоит в выявлении характерных особенностей, которые могут отличать эти два типа кода.

Задачи исследования включают:

- сбор выборки файлов человеческого и ИИ-кода;
- проведение предобработки данных для их сопоставимости;
- выделение параметров для анализа;
- сравнение распределений значений параметров в двух выборках;
- формулирование выводов о наиболее показательных различиях.

Данная работа отражает процесс и результаты анализа, направленного на понимание того, какие особенности могут быть использованы в дальнейшем

для разработки инструментов классификации кода по авторству. Несмотря на значительное число работ, посвящённых детекции ИИ-сгенерированного кода, большинство существующих подходов опирается либо на поверхностные стилевые признаки, либо на сложные нейросетевые представления, плохо поддающиеся интерпретации. В условиях образовательных, юридических и корпоративных сценариев это ограничивает практическую применимость детекторов. В настоящей работе предлагается иной подход, основанный на системе компактных, интерпретируемых метрик, отражающих структурно-семантические особенности программного кода.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Объект исследования — программный код на языке Python, используемый в учебных и прикладных задачах.

Предмет исследования — структурные, лексические и семантические характеристики программного кода, позволяющие различать код, написанный человеком, и код, сгенерированный искусственным интеллектом.

Цель магистерской диссертации — разработка и экспериментальная оценка метода бинарной классификации происхождения программного кода на основе интерпретируемых структурно-семантических метрик и классических алгоритмов машинного обучения.

Для достижения поставленной цели в работе решались следующие задачи:

- Выполнение анализа современных исследований в области детекции ИИ-сгенерированного программного кода.
- Формирование набора интерпретируемых метрик, отражающих структурные, лексические и семантические особенности программ.
- Разработка формальной постановки задачи бинарной классификации происхождения кода.
- Реализация экспериментальной методики обучения и оценки классификаторов.
- Проведение сравнительного анализа моделей Logistic Regression и Random Forest.
- Оценка качества классификации с использованием стандартных метрик и ROC-анализа.
- Выполнение анализа важности признаков и интерпретацию полученных результатов.

Методы исследования включают методы статического анализа программного кода, теорию машинного обучения, методы бинарной классификации, статистические методы оценки качества моделей, а также экспериментальное моделирование.

Научная новизна работы заключается в следующем:

- предложен компактный набор интерпретируемых структурно-семантических метрик для задачи детекции происхождения программного кода;
- показана эффективность классических методов машинного обучения без использования тяжёлых нейросетевых детекторов;
- выполнен сравнительный анализ линейных и ансамблевых моделей с точки зрения качества классификации и интерпретируемости признаков.

Практическая значимость работы заключается в возможности применения разработанного подхода в образовательных системах, при автоматической проверке учебных заданий, а также в экспертных системах анализа программного кода.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Магистерская диссертация состоит из введения, трёх глав, заключения, списка использованных источников и приложений.

В первой главе рассматриваются теоретические основы анализа программного кода и современные подходы к генерации программ с использованием искусственного интеллекта. Выполнен обзор методов статического и динамического анализа кода, интеллектуального анализа программ, а также существующих работ по детекции ИИ-сгенерированного кода. Особое внимание уделено ограничениям существующих подходов, таким как зависимость от длины файла, утечка стилевых признаков и слабая интерпретируемость нейросетевых моделей.

Во второй главе выполнена формальная постановка задачи бинарной классификации происхождения программного кода. Описано формирование признакового пространства на основе 13 метрик, включающих структурные, лексические и семантические характеристики программ. Введено математическое описание обучающей выборки, классов и векторов признаков. Также приведено описание используемых классификаторов и методики экспериментального исследования.

В третьей главе представлены результаты экспериментальной оценки предложенного подхода. Проведено обучение и тестирование моделей Logistic Regression и Random Forest. Качество классификации оценивалось с использованием метрик Accuracy, Precision, Recall, F1-score и ROC-AUC. Построены ROC-кривые, выполнен анализ важности признаков и проведено сравнение моделей. Показано, что модель Random Forest обеспечивает более высокое качество классификации за счёт способности учитывать нелинейные зависимости между признаками.

ЗАКЛЮЧЕНИЕ

В настоящей работе была исследована проблема автоматического определения происхождения программного кода – написан ли он человеком или сгенерирован нейросетевой моделью. Данная задача приобретает всё большую актуальность в связи с широким распространением генеративных инструментов программирования, которые активно используются как в профессиональной среде, так и в учебных целях. Рост доли искусственно созданного кода приводит к необходимости разработки надёжных методов его идентификации, что имеет важное значение для обеспечения безопасности, контроля качества, защиты интеллектуальной собственности и объективной оценки компетенций обучающихся.

В ходе исследования был сформирован репрезентативный датасет, включающий 1664 файла исходного кода на языке Python, из которых 799 написаны разработчиками, а 865 – сгенерированы современными моделями искусственного интеллекта (GPT-4o и o3-mini-high). Проведена многоступенчатая очистка данных, нормализация формата, удаление дубликатов, фильтрация служебных и тестовых файлов, а также проверка синтаксической корректности. Такой подход обеспечил высокое качество исходного материала и сформировал однородную выборку для последующего анализа.

Ключевым научным результатом стало создание комплексной системы из 13 метрик, отражающих структурные, лексические и семантико-контекстные особенности программ. Эти метрики охватывают как поверхностные количественные характеристики (длина кода, число функций, особенности импортов), так и более глубокие элементы стиля (тип предикатов, семантическое разнообразие вызовов, плотность исключений).

При вычислении признакового пространства каждая из 13 метрик была разложена в набор элементарных признаков (например, keyword frequency → 10 признаков, распределение предикатов → 5 признаков), что сформировало 29-мерный вектор признаков для каждой программы.

Экспериментальная часть исследования включала извлечение признаков, формирование обучающих и тестовых подвыборок с учётом принадлежности файлов проектам, а также обучение двух моделей машинного обучения: ансамблевой модели (Random Forest) и логистической регрессии. Оба классификатора продемонстрировали высокую точность:

- Random Forest: Accuracy = 0.944, ROC-AUC = 0.988 на тестовой выборке;

- Logistic Regression (с нормализацией): Accuracy = 0.918, ROC-AUC = 0.931.

Использование расширенного набора признаков позволило достичь высокой степени разделимости классов, а сама структура признаков продемонстрировала устойчивость к переобучению благодаря групповому разбиению по проектам.

Анализ важности признаков показал, что наибольший вклад в классификацию вносят:

- структурные характеристики (LOC, непустые строки, количество импортов);
- лексические признаки (частоты ключевых слов: class, import, def, return, for);
- семантико-контекстные признаки (semantic_call_diversity, predicate_type_distribution, eh_density).

Эти результаты подтверждают, что различия между человеческим и ИИ-кодом проявляются как в макроструктуре программы, так и в характере использования языковых конструкций и API.

Проведённый анализ ограничений работы показал влияние особенностей датасета, ограниченность семантических метрик, зависимости от синтаксиса Python и сложности интерпретации некоторых стилистических признаков. Тем не менее устойчивость результатов, воспроизводимость экспериментов и согласованность поведения обеих моделей подтверждают надёжность выводов исследования.

Перспективы дальнейшей работы включают:

- расширение набора языков программирования и адаптацию метрик под другие синтаксические семейства;
- применение бустинговых моделей и нейросетевых архитектур для анализа AST и CFG-структур;
- изучение архитектурных характеристик проектов (межфайловые зависимости, структура пакетов);
- разработку гибридных моделей, объединяющих статические метрики и эмбеддинги LLM;
- интеграцию разработанного метода в системы контроля качества кода и образовательные платформы.

Проведённое исследование позволяет сделать вывод о том, что предложенная система метрик и разработанная методика анализа являются эффективной и перспективной основой для распознавания происхождения программного кода. Полученные результаты подтверждают практическую значимость работы и открывают возможности для дальнейшего развития

подходов к анализу стиля, структуры и семантических свойств программ, создаваемых человеком или искусственным интеллектом.

Кроме того, выявленные закономерности демонстрируют, что даже относительно простые модели машинного обучения способны достоверно классифицировать происхождение кода при условии корректного подбора метрик и чистоты выборки. Это подчёркивает важность разработанной методики предобработки данных и подтверждает, что анализ структурных и семантических признаков может служить надёжным инструментом для задач аудита, образовательного контроля и обеспечения доверия к программным системам в условиях растущего использования генеративных технологий.

СПИСОК ПУБЛИКАЦИЯ СОИСКАТЕЛЯ

1 – Желтко, Ю. Ю. Влияние внедрения искусственного интеллекта в розничную торговлю / Ю. Ю. Желтко // Компьютерные системы и сети : сборник статей 60-й научной конференции аспирантов, магистрантов и студентов, Минск, 22–26 апреля 2024 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2024. – С. 44

1 – Желтко, Ю. Ю. Сравнительный анализ программного кода, написанного человеком и сгенерированного ИИ = Comparative analysis of human-written and ai-generated software code / Ю. Ю. Желтко // Компьютерные системы и сети : материалы 61-й научной конференции аспирантов, магистрантов и студентов, Минск, 22–26 апреля 2025 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2025. – С. 49–52.

2 – Zhaltko, Y. Validation of Parameters for AI Source Code Detector / E. Alooeff, Y. Zhaltko // Artificial Intelligence and Information Technologies : proceedings of the XV International Conference on Applied Internet and Information Technologies (AIIT 2025), Bitola, North Macedonia, 7 November 2025. – 2025. – P. 428-434.