

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНОГО КОДА, НАПИСАННОГО ЧЕЛОВЕКОМ И СГЕНЕРИРОВАННОГО ИИ

Желтко Ю.Ю., магистрант гр.355841

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

Одинец Д.Н. – канд. тех. наук

Аннотация. Исследование посвящено сравнению кода, написанного человеком и сгенерированного искусственным интеллектом. Производится сравнительный анализ этих типов кода на тестовых данных. Анализируются длина кода, количество комментариев и длина функций.

Ключевые слова. Искусственный интеллект (ИИ), анализ кода, длина функций, комментарии, структура программ.

Современное программное обеспечение все чаще создается с использованием инструментов на основе искусственного интеллекта (ИИ), включая генерацию кода. Это приводит к необходимости изучения различий между кодом, написанным человеком (КНЧ) и кодом, созданным нейросетевыми моделями (КСНМ). Целью данного исследования является выявление характерных особенностей обоих типов кода, определение различий в длине кода, длине функций и количестве комментариев. Данный анализ необходим не только для понимания технических аспектов программного обеспечения, но и для оценки перспектив дальнейшей автоматизации разработки ПО с помощью ИИ.

Подобные исследования, посвященные анализу программного кода, изучающее различия между КНЧ и КСНМ имеет ряд аналогов. Например, в исследовании «Evaluating Code Readability and Legibility: An Examination of Human-centric Studies» [1] анализируется влияние таких факторов как используемые программные конструкции и соглашения об именах, влияют на читаемость и разборчивость кода. В исследовании «Between Lines of Code: Unraveling the Distinct Patterns of Machine and Human Programmers» [2] выявляются различия КНЧ и КСНМ. Авторы анализируют такие характеристики, как лексическое разнообразие, краткость и естественность кода, выявляя уникальные паттерны для каждого источника. Исследование демонстрирует, что машинный код часто обладает уникальными особенностями, такими как повторяемость шаблонов и нестандартное использование операторов.

Исследование рассматриваемых видов кода производилось по следующим параметрам:

1. Длина кода: Измерение распределения длины кода в различных файлах с целью определения, насколько различаются объемы кода, написанного человеком и ИИ.

2. Комментарии и документирование: Оценка доли закомментированных строк в коде с целью определения уровня документированности и сопровождаемости.

3. Длина функций: Анализ средней и максимальной длины функций для выявления различий в структурировании кода и предпочтениях при написании логических блоков.

Эти характеристики напрямую влияют на качество, сопровождаемость и надёжность программного обеспечения.

Исследование проводилось на наборе из 1000 файлов КНЧ и более 900 файлов КСНМ. Данные были собраны из открытых репозиториях сервиса Github.com, созданных до 2019 года, не редактированных позднее 2020 года и структурированы для интеллектуального анализа. Для генерации кода ИИ был использован сервис ChatGPT с моделями 4o и o3-mini-high. Выбор именно этих моделей обусловлен их широким распространением, открытостью использования и высоким качеством генерации кода, подтверждённым многочисленными тестами и отзывами разработчиков [3].

На рисунке 1 представлено распределение длины кода относительно количества исследуемых файлов для двух типов данных: КНЧ и КСНМ. Анализ показал, что КНЧ существенно длиннее и сложнее, чем КСНМ. На диаграмме КНЧ длина файлов варьируется от 0 до более чем 6000 строк. В КСНМ диапазон значительно меньше – от 0 до 400 строк. Данное распределение свидетельствует о том, что ИИ склонен генерировать короткие и компактные программы. В КСНМ наибольшее количество файлов сосредоточено в диапазоне от 0 до 200 строк, и пик приходится на 50-100 строк. Полученные результаты позволяют предположить, что нейросетевые модели чаще используют стандартизированные подходы к решению задач, избегая сложных или многоуровневых алгоритмических конструкций. Это может объясняться как ограничениями текущих технологий, так и спецификой обучающих данных, на которых тренируются модели ИИ.

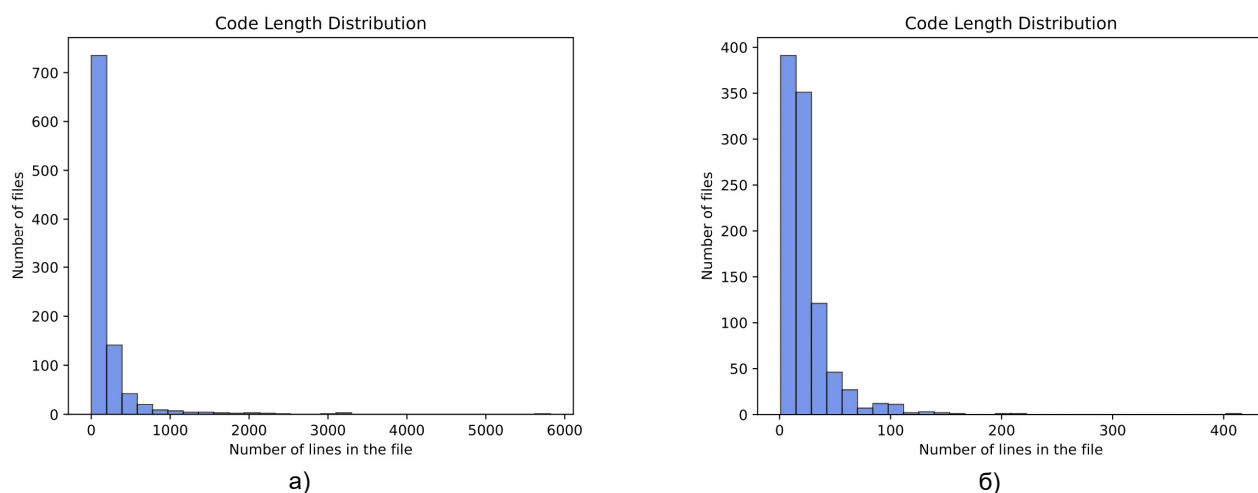


Рисунок 1 – Распределение длины кода: а – написанного человеком; б – написанного ИИ

На рисунке 2 представлены распределения комментариев, относительно числу строк в файлах.

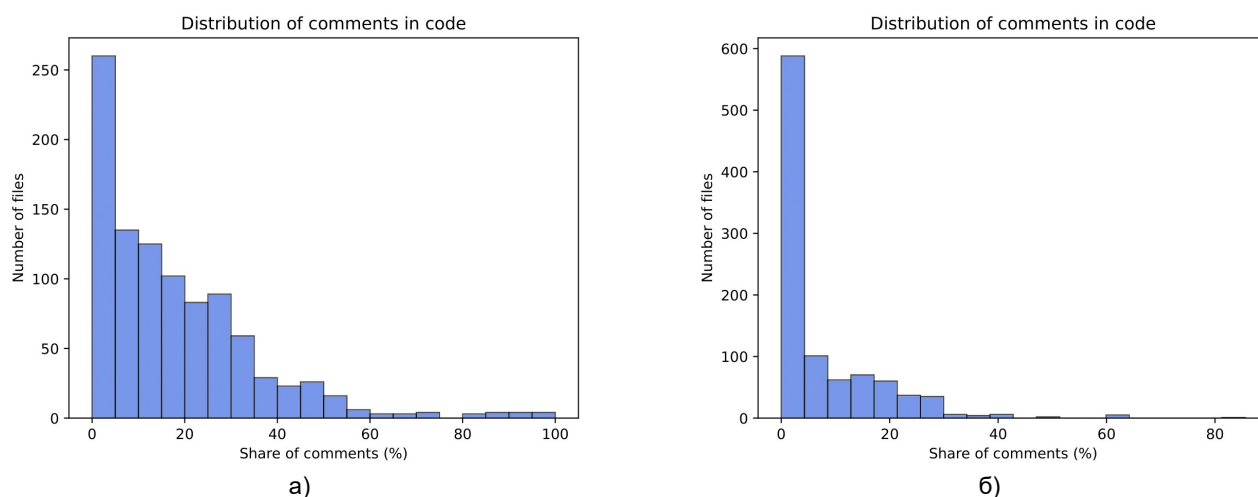


Рисунок 2 – Распределение комментариев в коде: а – написанного человеком; б – написанного ИИ

Комментарии играют ключевую роль в понимании и сопровождении кода. Они помогают разработчикам быстро ориентироваться в логике программы и облегчают процесс обновления и модификации кода. Отсутствие или несоответствие комментариев может привести к недопониманию и увеличению количества ошибок. Исследования [4] показывают, что несогласованность между кодом и комментариями увеличивает вероятность внесения ошибок в 1,5 раза. Поэтому качественные и актуальные комментарии являются неотъемлемой частью надёжного программного обеспечения.

Анализ показал, что комментирование КНЧ значительно более разнообразно и может достигать очень высокой доли (до 100%), что часто встречается в учебных и академических проектах. КСНМ почти всегда имеет низкую долю комментариев (до 20%), что говорит о слабой способности ИИ к генерации объяснительного текста. КНЧ демонстрирует как низкую, так и высокую насыщенность комментариями, тогда как КСНМ практически лишён файлов с высокой долей комментариев.

На рисунке 3 представлены распределения длины функций относительно их количества. Длина функций оказывает существенное влияние на читаемость и сопровождаемость кода. Функции, содержащие более 143 строк, требуют в 2,4 раза больше усилий на исправление ошибок по сравнению с более короткими функциями [5]. Кроме того, высокая цикломатическая сложность, связанная с длинными функциями, увеличивает риск появления ошибок и затрудняет тестирование.

Анализ показал, что функции в КНЧ существенно длиннее и разнообразнее по структуре, чем функции, созданные ИИ. На диаграмме КНЧ длина функций варьируется от 0 до более 70 строк. В КСНМ диапазон значительно меньше – от 0 до примерно 50 строк. Данное распределение свидетельствует о том, что ИИ склонен генерировать более короткие функции. В КНЧ наибольшее количество функций сосредоточено в диапазоне от 5 до 15 строк, и пик приходится на 10 строк. В КСНМ также наблюдается пик в диапазоне 5-15 строк, но общее количество функций гораздо меньше. Это указывает на ограниченность структуры и сложности функций, генерируемых ИИ, по сравнению с КНЧ.

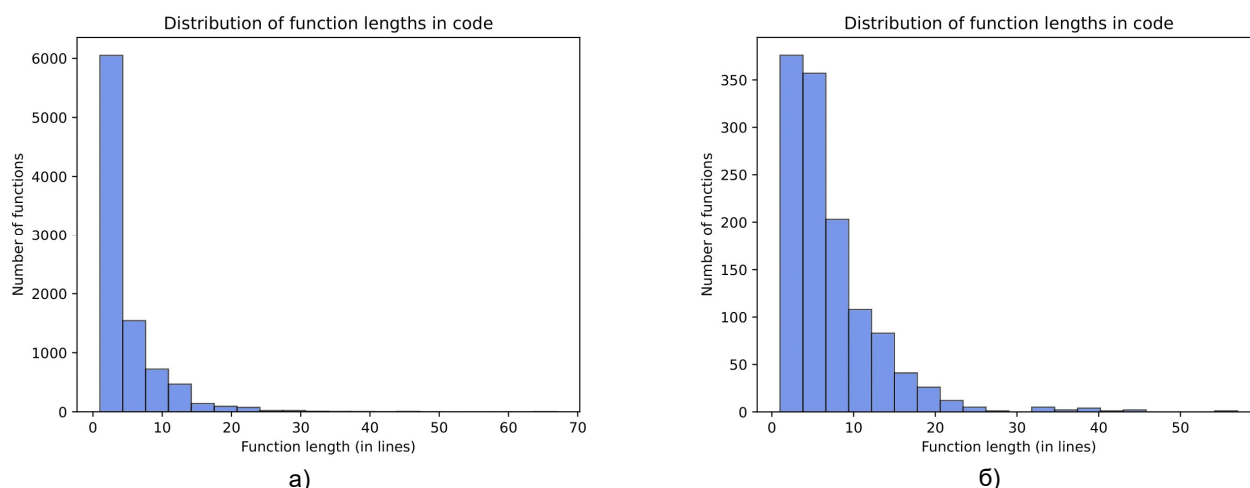


Рисунок 3 – Распределение длины функций в коде: а – написанного человеком; б – написанного ИИ

Исследование показало, что КСНМ имеет более высокую читаемость и меньшую длину, но обладает недостаточной степенью комментирования и глубиной вложенности логических конструкций. КНЧ, несмотря на большую сложность и разнообразие, демонстрирует гибкость и способность решать более сложные задачи за счёт глубокой проработки архитектуры. Обнаруженные различия между КНЧ и КСНМ позволяют более критически оценивать результаты работы генеративных моделей, а также помогут разработчикам и исследователям сформулировать рекомендации по использованию подобных технологий в реальных проектах

Полученные данные могут быть полезны:

1. В образовательных платформах. Данные из настоящего исследования могут быть полезны для разработки методических рекомендаций, направленных на улучшение структуры кода в обучающих средах.
2. В системах автоматического анализа кода: полученные данные могут использоваться для улучшения статических анализаторов.
3. Для научных исследований: результаты создают основу для дальнейшего сравнения кода, написанного человеком, с кодом, сгенерированным нейросетями. Полученные в данном исследовании данные могут быть использованы для более точного анализа и классификации кода по его происхождению.

Таким образом, результаты исследования демонстрируют, что нейросетевые генераторы кода эффективны для решения простых и шаблонных задач, но при создании сложных и многоуровневых программ требуется профессиональная разработка с учётом архитектурных и концептуальных особенностей кода.

Список использованных источников:

1. *Evaluating Code Readability and Legibility: An Examination of Human-centric Studies* [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2110.00785> – Дата доступа: 04.04.2025.
2. *Between Lines of Code: Unraveling the Distinct Patterns of Machine and Human Programmers* [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2401.06461> – Дата доступа: 04.04.2025.
3. *ChatGPT 4o vs o3-mini: OpenAI's Next-Generation AI Models* [Электронный ресурс]. – Режим доступа: <https://www.geeksforgeeks.org/chatgpt-4o-vs-o3-mini/> – Дата доступа: 14.04.2025.
4. *Investigating the Impact of Code Comment Inconsistency on Bug Introducing* [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2409.10781> – Дата доступа: 04.04.2025.
5. *The Impact of Code Readability on Software Maintenance efficiency in open source development* [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/390151582_The_Impact_of_Code_Readability_on_Software_Maintenance_efficiency_in_open_source_development – Дата доступа: 04.04.2025.

COMPARATIVE ANALYSIS OF HUMAN-WRITTEN AND AI-GENERATED SOFTWARE CODE

Zhaltko Y.Y.

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

Adzinets D.N. – PhD in Technical science

Annotation. The study is devoted to comparative code written by a person and generated by artificial intelligence. A comparative analysis of this type of code is performed on test data. The code length, number of comments and length of functions are analyzed.

Keywords. Artificial Intelligence, ai code analysis, function length, comments, program structure.