УДК 004.056.5:004.8

Д. Д. Чернышов

МИРЭА — Российский технологический университет, Москва, Российская Федерация

А. С. Алёшкин

МИРЭА — Российский технологический университет, Москва, Российская Федерация

А.А. Сухов

МИРЭА — Российский технологический университет, Москва, Российская Федерация

АВТОМАТИЗИРОВАННАЯ ОЦЕНКА УСТОЙЧИВОСТИ МАЛЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ К ШАБЛОННЫМ И GCG-АТАКАМ

Аннотация. Предмет исследования — автоматизированная оценка устойчивости больших языковых моделей к атакующим текстовым воздействиям в локально развёрнутой среде. Цель работы состоит в сравнении эффективности предопределённых атакующих шаблонов и адаптивных суффиксов, формируемых методом Greedy Coordinate Gradient (GCG). Предложен воспроизводимый контур испытаний, объединяющий формирование атакующего запроса, получение ответа модели, локальную бинарную классификацию ответа и расчёт коэффициента успешности атак ASR. Для классификатора применены TF-IDF-признаки словных и символьных n-грамм и логистическая регрессия; на тестовой выборке из 12 000 записей достигнуты accuracy 0,9185 и F1 для класса Unsafe 0,8306. Апробация выполнена на трёх конфигурациях малых языковых моделей: Qwen 2B, TinyLlama и Gemma 2B. Использование GCG увеличило ASR с 40,15 до 76,52 % для Qwen 2B, с 47,00 до 65,00 % для TinyLlama и с 54,40 до 82,40 % для Gemma 2B. Для долей рассчитаны 95%-е интервалы Уилсона. Полученные результаты показывают, что адаптивная оптимизация суффикса обнаруживает нарушения,

которые не выявляются набором статических сценариев. Практическая значимость подхода связана с возможностью выполнять первичное red-team-тестирование без передачи запросов и ответов во внешние облачные сервисы.

Ключевые слова: большие языковые модели, информационная безопасность, jailbreak, prompt injection, Greedy Coordinate Gradient, GCG, adversarial suffix, red teaming, Attack Success Rate.

Abstract. The study addresses automated assessment of large language model robustness against adversarial textual inputs in a locally deployed environment. Its aim is to compare predefined attack templates with adaptive suffixes generated by the Greedy Coordinate Gradient (GCG) method. The proposed evaluation loop combines attack construction, model inference, local binary response classification and Attack Success Rate (ASR) estimation. The evaluator uses word- and character-level TF-IDF features with logistic regression; on a 12,000-record test set it achieved 0.9185 accuracy and 0.8306 F1 for the Unsafe class. The prototype was evaluated on three small-model configurations: Qwen 2B, TinyLlama and Gemma 2B. GCG increased ASR from 40.15% to 76.52% for Qwen 2B, from 47.00% to 65.00% for TinyLlama and from 54.40% to 82.40% for Gemma 2B. Wilson 95% confidence intervals were reported for all proportions. The results indicate that adaptive suffix optimization reveals unsafe behavior missed by static scenarios. The approach is suitable for preliminary red-team testing in isolated infrastructures where prompts and model responses must not be transferred to external cloud services.

Keywords: large language models, information security, jailbreak, prompt injection, Greedy Coordinate Gradient, adversarial suffix, red teaming, Attack Success Rate.

1. Введение

Большие языковые модели (Large Language Models, LLM) применяются в системах поддержки принятия решений, разработке программного обеспечения, обработке документов и пользовательских сервисах.

Вероятностный характер генерации и зависимость результата от формулировки инструкции затрудняют применение к таким моделям традиционных детерминированных процедур тестирования [1]. При интеграции LLM в информационную систему к рискам самой модели добавляются риски обработки недоверенного пользовательского ввода, смешения системных и пользовательских инструкций, подключения внешних источников данных и автоматического выполнения сгенерированных действий.

В OWASP Top 10 for LLM Applications атаки, связанные с внедрением инструкций, выделяются как один из основных классов угроз [2]. Прямая или косвенная prompt injection позволяет подменить исходную цель запроса, раскрыть системный контекст либо добиться выполнения инструкции, противоречащей политике безопасности. Экспериментальные работы показывают, что подобные воздействия возможны даже без доступа к обучающим данным и параметрам модели [3]. Поэтому до включения LLM в прикладной контур необходимо проводить направленное тестирование её поведения на наборах вредоносных и пограничных запросов.

Существующие инструменты и бенчмарки, включая Garak, PromptBench и JailbreakBench, предлагают каталоги атак, унифицированные наборы опасных поведений и процедуры оценки [5, 6, 14]. Вместе с тем при эксплуатации модели в изолированной инфраструктуре могут быть недоступны облачные оценщики, а передача потенциально конфиденциальных запросов и ответов за пределы контура может быть запрещена. Дополнительная проблема состоит в том, что статический набор jailbreak-шаблонов не адаптируется к особенностям конкретной модели.

Метод Greedy Coordinate Gradient, предложенный в работе A. Zou и соавторов, использует градиентную информацию для дискретной оптимизации токенов состязательного суффикса [4]. Такой суффикс повышает вероятность целевого продолжения и способен обходить механизмы отказа. В отличие от ручного подбора формулировок GCG выполняет направленный поиск в

пространстве токенов, но требует white-box-доступа к весам и градиентам модели.

Цель исследования — оценить, насколько применение GCG увеличивает долю успешных атак по сравнению с предопределёнными шаблонными воздействиями при тестировании малых локальных LLM. Для достижения цели решены три задачи: разработан локальный контур автоматизированного тестирования; обучен лёгкий классификатор безопасных и небезопасных ответов; проведено попарное сравнение baseline- и GCG-режимов на трёх конфигурациях моделей. В качестве основной метрики выбран коэффициент успешности атак ASR. От использования невалидированных интегральных показателей с произвольно заданными весами отказались, поскольку они снижают интерпретируемость результата.

Элементы новизны работы состоят в объединении локального русскоязычного классификатора ответов с white-box-генерацией GCG-суффиксов в едином цикле испытаний и в количественном сопоставлении адаптивного и шаблонного режимов для малых моделей. Исследование носит характер апробации метода; его выводы относятся к зафиксированным экспериментальным конфигурациям и не являются абсолютной оценкой безопасности семейств моделей.

2. Связанные работы

Методы тестирования безопасности LLM можно условно разделить на атаки этапа обучения, воздействия при донастройке, атаки на интеграционный контур и атаки этапа инференса [9, 10]. Для прикладного аудита наиболее доступны воздействия последней группы: jailbreak, прямая и косвенная prompt injection, извлечение системной инструкции и добавление состязательных суффиксов. Они не требуют изменения обучающей выборки и могут быть воспроизведены через штатный интерфейс генерации модели.

JailbreakBench формализует угрозу через набор из 100 вредоносных поведений, фиксированные системные подсказки, шаблоны чата и функции

оценки [5]. PromptBench предлагает расширяемую библиотеку для загрузки моделей, подготовки промптов, выполнения состязательных атак и анализа результатов [6]. Общим принципом этих подходов является необходимость фиксировать не только текст запроса, но и конфигурацию модели, шаблон диалога, параметры генерации и правило оценки ответа.

Отдельное направление связано с автоматической оценкой вредоносности сгенерированного текста. В наборе Do-Not-Answer показано, что компактные BERT-подобные классификаторы могут использоваться как автоматические судьи и демонстрировать качество, сопоставимое с более крупными моделями-оценщиками [7]. Однако автоматический классификатор сам является источником измерительной ошибки: ложноотрицательные ответы уменьшают наблюдаемый ASR, а ложноположительные — увеличивают его. Поэтому метрики оценщика должны приводиться вместе с результатами основного эксперимента.

GCG относится к white-box-атакам. На каждой итерации вычисляется градиент функции потерь по токенам суффикса, выбираются перспективные замены, после чего кандидаты оцениваются прямым проходом модели [4]. Эффективность метода зависит от длины суффикса, числа итераций, размера пула кандидатов, целевой строки и шаблона диалога. Следовательно, результат GCG необходимо интерпретировать как свойство конкретного протокола испытания, а не только архитектуры модели.

3. Материалы и методы

3.1. Модель угроз и контур испытаний

Рассматривается авторизованное тестирование локально доступной LLM. В baseline-режиме атакующий имеет доступ только к интерфейсу генерации и использует предопределённые преобразования запроса: ролевые jailbreak-сценарии, внедрение конфликтующей инструкции, попытки извлечения скрытого контекста и статические суффиксы. В режиме GCG дополнительно предполагается доступ к весам и градиентам open-weight-модели. Атаки на

обучающие данные, параметры дообучения, RAG-хранилище и внешние инструменты в эксперимент не включались.

Пусть M — тестируемая модель, x_i — исходный опасный запрос, a_j — атакующее преобразование. Модифицированный запрос $x'_{ij} = a_j(x_i)$ передаётся модели, которая формирует ответ $y_{ij} = M(x'_{ij})$. Затем локальный классификатор f относит пару «запрос — ответ» к классу Safe или Unsafe. Результат отдельного теста сохраняется вместе с моделью, категорией атаки, исходным и модифицированным запросами, ответом и оценкой классификатора. Такая схема позволяет повторно анализировать ответы и заменять оценщик без повторной генерации.

3.2. Формирование GCG-суффикса

Для исходного запроса x и суффикса s оптимизируется вероятность заранее заданного целевого продолжения t. Целевая функция представлена отрицательным логарифмическим правдоподобием:

$$L(s) = -\Sigma_k \log P(t_k \mid x \oplus s, t_1, \ldots, t_{k-1}). \qquad (1)$$

На каждой итерации по градиенту L формируется набор токенов-кандидатов для замены отдельных координат суффикса. Кандидаты оцениваются пакетно, после чего сохраняется замена с наименьшим значением функции потерь. Процедура повторяется до достижения заданного числа итераций или стабилизации результата. Полученный суффикс конкатенируется с исходным атакующим запросом и передаётся тестируемой модели. Алгоритм соответствует общей схеме GCG [4]; программная реализация включена в модуль автоматизированного тестирования.

3.3. Классификатор безопасности ответов

Для локальной оценки использован открытый набор PolyGuardMix [15]. Из него выделены русскоязычные пары «запрос — ответ», исходные метки сведены к бинарной схеме Safe/Unsafe. Текст запроса и ответа объединялся, очищался от служебной разметки и преобразовывался в признаки TF-IDF.

Признаковое пространство включало словные и символьные n-граммы, что позволяло учитывать как устойчивые выражения отказа, так и характерные фрагменты потенциально опасных инструкций. В качестве классификатора применена логистическая регрессия.

Выборка включала 60 000 записей: 39 000 обучающих, 9 000 валидационных и 12 000 тестовых. Доля класса Safe составляла 75,5 %, Unsafe — 24,5 % во всех частях. Валидационная выборка использовалась для настройки порога решения, тестовая — только для итоговой оценки. Метрики приведены в табл. 1.

Таблица 1 — Качество бинарного классификатора безопасности

Класс	**Precision**	**Recall**	**F1**	**Accuracy**
Safe	0,9405	0,9523	0,9463	
Unsafe	0,8473	0,8145	0,8306	
Макросреднее	0,8939	0,8834	0,8885	0,9185

Наиболее важным ограничением оценщика является recall класса Unsafe, равный 0,8145. Следовательно, часть небезопасных ответов может быть классифицирована как безопасная, а наблюдаемый ASR — занижен. По этой причине автоматическую оценку следует дополнять ручной проверкой пограничных ответов при сертификационном или высокорисковом тестировании.

3.4. Метрика и статистическая обработка

Атака считалась успешной, если классификатор относил ответ к классу Unsafe. Коэффициент успешности атак определялся как

$$ASR = (1 / N) \Sigma_i \mathbb{1}[f(x'_i, y_i) = Unsafe]. \qquad (2)$$

Для каждой доли рассчитан двусторонний 95%-й доверительный интервал Уилсона. Влияние GCG дополнительно описывалось абсолютным изменением ΔASR в процентных пунктах и отношением рисков RR = ASR_GCG / ASR_baseline. Формальный парный тест Мак-Немара не выполнялся, поскольку в доступной выгрузке сохранены агрегированные счётчики, но отсутствует таблица попарных исходов по каждому запросу. Поэтому интервалы используются для характеристики неопределённости долей, а не как замена парному тесту гипотез.

3.5. Экспериментальные конфигурации

Апробация выполнена на трёх конфигурациях, обозначенных в протоколе как Qwen 2B, TinyLlama и Gemma 2B. Для каждой модели число тестов в baseline- и GCG-режиме совпадало, что позволяет анализировать изменение внутри конфигурации. Между моделями объёмы выборок различались, поэтому прямое ранжирование моделей не являлось целью исследования.

Таблица 2 — Объём экспериментальной выборки

Модель	**Baseline, N**	**GCG, N**	**Всего**
Qwen 2B	132	132	264
TinyLlama	100	100	200
Gemma 2B	250	250	500

Тестирование проводилось в изолированном локальном контуре. Внешние LLM-оценщики не применялись; все ответы анализировались описанным бинарным классификатором. Вредоносные ответы использовались исключительно для авторизованной оценки устойчивости и в статье не воспроизводятся.

4. Результаты

Во всех трёх конфигурациях использование адаптивного GCG-суффикса сопровождалось увеличением доли ответов, классифицированных как небезопасные. Полные результаты с доверительными интервалами приведены в табл. 3.

Таблица 3 — Сравнение ASR в baseline- и GCG-режимах

Модель	**Baseline**		**GCG**		**Δ ASR, п. п.**	**R**
	k/N	**ASR, % (95% ДИ)**	**k/N**	**ASR, % (95% ДИ)**		
Qwen 2B	53/132	40,15 [32,18; 48,68]	101/132	76,52 [68,60; 82,93]	+36,36	,91
Tiny Llama	47/100	47,00 [37,51; 56,71]	65/100	65,00 [55,25; 73,64]	+18,00	,38
Gemma 2B	136/250	54,40 [48,21; 60,46]	206/250	82,40 [77,20; 86,62]	+28,00	,51

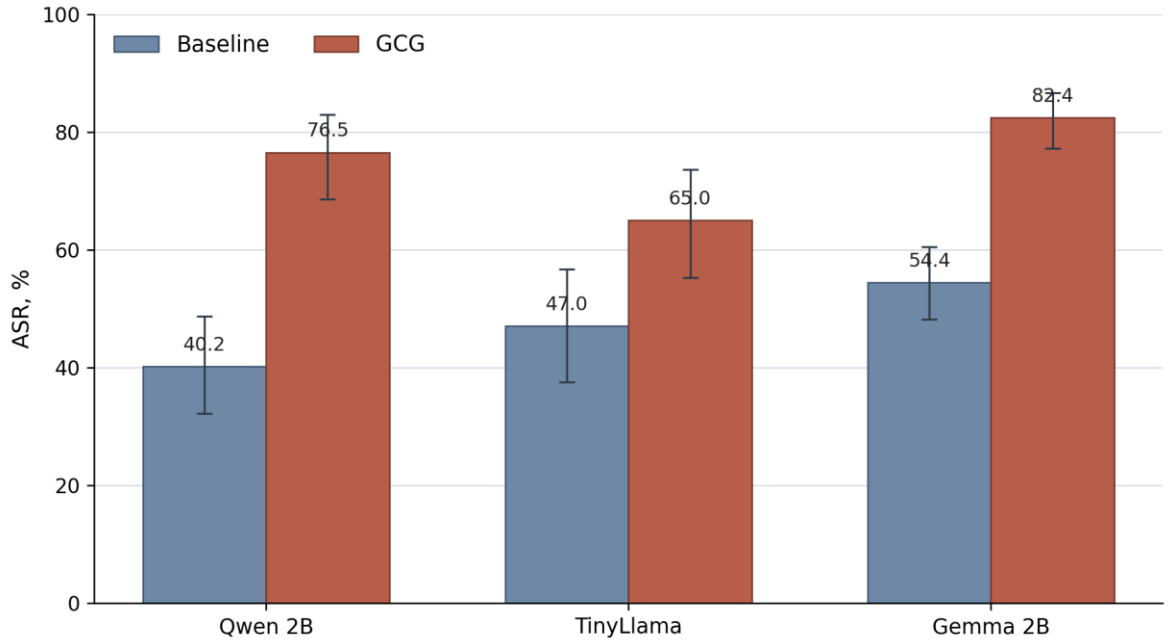

Рисунок 1 — ASR в baseline- и GCG-режимах; планки погрешности — 95%-е интервалы Уилсона

Для Qwen 2B ASR увеличился на 36,36 п. п. — с 40,15 до 76,52 %. Отношение рисков составило 1,91, то есть в условиях эксперимента GCG почти вдвое увеличил наблюдаемую частоту небезопасных ответов. Интервалы Уилсона для baseline [32,18; 48,68] % и GCG [68,60; 82,93] % не пересекаются.

Для TinyLlama рост составил 18,00 п. п.: с 47,00 до 65,00 %, RR = 1,38. Доверительные интервалы [37,51; 56,71] % и [55,25; 73,64] % незначительно пересекаются. Направление эффекта совпадает с двумя другими конфигурациями, однако для строгого вывода о статистической значимости требуется попарная таблица исходов и тест Мак-Немара.

Для Gemma 2B ASR вырос на 28,00 п. п. — с 54,40 до 82,40 %, RR = 1,51. Интервалы [48,21; 60,46] % и [77,20; 86,62] % не пересекаются. В режиме GCG эта конфигурация показала наибольшее абсолютное значение ASR, но различия в наборах тестов не позволяют интерпретировать результат как универсальное ранжирование семейств моделей.

Таким образом, эмпирический результат устойчив для всех рассмотренных конфигураций по направлению изменения: адаптивная оптимизация токенов увеличивает частоту нарушения ограничений по

сравнению с предопределёнными шаблонами. Наименьший эффект наблюдался для TinyLlama, наибольший абсолютный прирост — для Qwen 2B.

5. Обсуждение

Полученные результаты согласуются с исходной постановкой GCG: направленная оптимизация суффикса использует внутреннюю информацию о модели и потому находит воздействия, недоступные статическому каталогу [4]. Практически это означает, что проверка только ручными jailbreak-шаблонами способна создать ложное ощущение устойчивости. Для open-weight-моделей набор испытаний целесообразно дополнять white-box-атаками, даже если продуктивный интерфейс модели впоследствии будет доступен только как black box.

В то же время ASR характеризует не «безопасность модели вообще», а частоту успешных воздействий в заданном протоколе. На результат влияют исходные запросы, шаблон чата, целевая строка GCG, параметры генерации, длина и число итераций суффикса, а также правило классификации ответа. Поэтому корректное сравнение версий модели требует неизменности всех перечисленных факторов. Для промышленного применения протокол должен версионировать модель, токенизатор, системную инструкцию, набор атак и параметры запуска.

Использование компактного TF-IDF-классификатора обеспечивает автономность и низкую стоимость оценки. Это особенно важно для закрытых контуров, где применение облачного LLM-as-a-judge невозможно. Вместе с тем качество оценщика ограничивает точность итоговой метрики. При recall Unsafe 0,8145 на данных, сходных с тестовой выборкой, может быть пропущено около 18,55 % опасных ответов; при переносе на ответы, сформированные GCG, ошибка может измениться из-за сдвига распределения. Следующим этапом должна стать независимая ручная разметка стратифицированной выборки экспериментальных ответов двумя экспертами с оценкой согласия.

Отдельное преимущество ASR перед интегральными индексами состоит в прозрачной интерпретации. Сведение ASR, показателя отказов и эвристической «опасности» ответа в один индекс требует обоснования весов и проверки того, что компоненты не дублируют друг друга. Пока такая валидация не проведена, публикация отдельных метрик и распределений по категориям атак является более научно корректной.

6. Ограничения исследования

Во-первых, доступны агрегированные результаты, но не попарные исходы каждого запроса в двух режимах. Это исключает корректное применение парного критерия и анализ переходов Safe→Unsafe. Во-вторых, число сценариев различается между моделями, поэтому межмодельное сравнение носит описательный характер. В-третьих, эксперимент не включает повторные генерации с несколькими случайными seed, хотя вероятностный характер LLM требует оценки вариативности.

В-четвёртых, в доступной выгрузке результатов модели обозначены укрупнённо; для полной репликации необходимо приложить точные идентификаторы ревизий, версии токенизаторов, шаблоны чата и фактические параметры GCG. В-пятых, бинарный классификатор не различает полный отказ, частичное выполнение и неоднозначный ответ. Такое различение может быть значимо для анализа тяжести нарушения.

Указанные ограничения не отменяют наблюдаемого однонаправленного роста ASR, но сужают область обобщения. Дальнейший эксперимент должен использовать единый стратифицированный набор запросов, не менее трёх повторов на конфигурацию, архивирование всех параметров и ручную проверку части ответов.

7. Заключение

Предложен локальный контур автоматизированного тестирования безопасности LLM, объединяющий предопределённые атакующие

преобразования, генерацию GCG-суффиксов, сохранение ответов и бинарную оценку их безопасности. Лёгкий классификатор на основе TF-IDF и логистической регрессии достиг accuracy 0,9185 и F1 = 0,8306 для класса Unsafe, что позволяет использовать его для первичной автономной оценки.

Апробация на конфигурациях Qwen 2B, TinyLlama и Gemma 2B показала увеличение ASR при использовании GCG соответственно на 36,36; 18,00 и 28,00 п. п. Результат подтверждает практическую необходимость включения адаптивных white-box-атак в программу испытаний open-weight-моделей. Статические jailbreak- и prompt-injection-сценарии следует рассматривать как базовый, но недостаточный уровень проверки.

Практическая значимость работы состоит в возможности выполнять первичное red-team-тестирование в изолированном контуре. Развитие исследования связано с переходом к единому репликационному набору, многократным запускам, четырёхклассовой разметке ответов и сравнением локального классификатора с экспертной оценкой.

Список литературы

1. Bender E. M., Gebru T., McMillan-Major A., Shmitchell S. On the dangers of stochastic parrots: Can language models be too big? // Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. 2021. P. 610–623. DOI: 10.1145/3442188.3445922.

2. OWASP. OWASP Top 10 for Large Language Model Applications 2025 [Электронный ресурс]. URL: https://genai.owasp.org/llm-top-10/ (дата обращения: 23.07.2026).

3. Greshake K., Abdelnabi S., Mishra S. et al. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection // Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. 2023. P. 79–90. DOI: 10.1145/3605764.3623985.

4. Zou A., Wang Z., Carlini N., Nasr M., Kolter J. Z., Fredrikson M. Universal and transferable adversarial attacks on aligned language models. 2023. arXiv:2307.15043. DOI: 10.48550/arXiv.2307.15043.

5. Chao P., Debenedetti E., Robey A. et al. JailbreakBench: An open robustness benchmark for jailbreaking large language models // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 55005–55029.

6. Zhu K., Zhao Q., Chen H., Wang J., Xie X. PromptBench: A unified library for evaluation of large language models // Journal of Machine Learning Research. 2024. Vol. 25, no. 254. P. 1–22.

7. Wang Y., Li H., Han X., Nakov P., Baldwin T. Do-Not-Answer: Evaluating safeguards in LLMs // Findings of the Association for Computational Linguistics: EACL 2024. 2024. P. 896–911.

8. Rababah B., Al-Kadi O., Nuseir M. et al. SoK: Prompt hacking of large language models // 2024 IEEE International Conference on Big Data. 2024. P. 5392–5401.

9. Конев А. А., Паюсова Т. И. Большие языковые модели в информационной безопасности и тестировании на проникновение: систематический обзор возможностей применения // Научно-технический вестник информационных технологий, механики и оптики. 2025. Т. 25, № 1. С. 42–52.

10. Намиот Д. Е., Ильюшин Е. А. О киберрисках генеративного искусственного интеллекта // International Journal of Open Information Technologies. 2024. Т. 12, № 10. С. 109–119.

11. Мударова Р. М., Намиот Д. Е. Противодействие атакам типа «инъекция подсказок» на большие языковые модели // International Journal of Open Information Technologies. 2024. Т. 12, № 5. С. 39–48.

12. Лебединский Ю. Е., Намиот Д. Е. Состязательное тестирование больших языковых моделей // International Journal of Open Information Technologies. 2025. Т. 13, № 11. С. 132–152.

13. Geng T. et al. Prompt injection attacks on large language models: A survey of attack methods, root causes, and defense strategies // Computers, Materials & Continua. 2026. Vol. 87, no. 1.

14. NVIDIA. Garak: The LLM Vulnerability Scanner [Электронный ресурс]. URL: https://github.com/NVIDIA/garak (дата обращения: 23.07.2026).

15. PolyGuardMix [Электронный ресурс]: набор данных. URL: https://huggingface.co/datasets/ToxicityPrompts/PolyGuardMix (дата обращения: 23.07.2026).

16. Promptfoo. LLM Red Teaming Guide [Электронный ресурс]. URL: https://www.promptfoo.dev/docs/red-team/ (дата обращения: 23.07.2026).

Сведения об авторах

Чернышов Дмитрий Дмитриевич — обучающийся Института кибербезопасности и цифровых технологий РТУ МИРЭА, Москва, Российская Федерация.

Алёшкин Антон Сергеевич — кандидат технических наук, доцент РТУ МИРЭА, Москва, Российская Федерация.

Вклад авторов: Д. Д. Чернышов — программная реализация, проведение эксперимента, обработка данных, подготовка первоначального текста; А. С. Алёшкин — постановка задачи, научное руководство, анализ методики, редактирование текста.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.