Программные системы и вычислительные методы
Правильная ссылка на статью:
Крохин А.С., Гусев М.М. Анализ влияния обфускации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73939 EDN: FBOXHC URL: https://nbpublish.com/library_read_article.php?id=73939

Анализ влияния обфускации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок

Крохин Алексей Сергеевич

студент; Московский институт электроники и математики; Национальный исследовательский университет "Высшая школа экономики"

109316, Россия, г. Москва, Таганский р-н, Волгоградский пр-кт, д. 13

askrokhin@edu.hse.ru

Гусев Максим Михайлович

студент; Московский институт электроники и математики; Национальный исследовательский университет "Высшая школа экономики"

109117, Россия, г. Москва, р-н Кузьминки, Волгоградский пр-кт, д. 115 к. 3

gusevmaxim04@mail.ru

Статья из рубрики "Модели и методы управления информационной безопасностью"

DOI:

10.7256/2454-0714.2025.2.73939

EDN:

FBOXHC

Дата направления статьи в редакцию:

02-04-2025

Дата публикации:

21-05-2025

Аннотация: В статье рассматривается проблема обфускации промптов как способа обхода защитных механизмов в больших языковых моделях (LLM), предназначенных для обнаружения промпт-инъекций. Промпт-инъекции представляют собой метод атаки, при

котором злоумышленники манипулируют входными данными, чтобы изменить поведение модели и заставить её выполнять нежелательные или вредоносные действия. Обфускация включает в себя различные методы изменения структуры и содержания текста, такие как замена слов синонимами, перемешивание букв в словах, вставка случайных символов и другие. Цель обфускации — затруднить анализ и классификацию текста, чтобы обойти фильтры и защитные механизмы, встроенные в языковые модели. В рамках исследования проводится анализ эффективности различных методов обфускации в обходе моделей, обученных на задачу классификации текста. Особое внимание уделяется оценке потенциальных последствий обфускации для безопасности и защиты данных. В исследовании используются различные методы обфускации текстов, которые применяются к промптам из датасета AdvBench. Эффективность методов оценивается на примере трёх моделей-классификаторов, обученных на задачу обнаружения промпт-инъекций. Научная новизна исследования заключается в анализе влияния обфускации промптов на эффективность языковых моделей в обнаружении промпт-инъекций. В ходе работы выявлено, что применение сложных методов обфускации увеличивает долю запросов, классифицируемых как инъекции, что подчёркивает необходимость тщательного подхода к тестированию безопасности больших языковых моделей. Выводы исследования указывают на важность баланса между сложностью метода обфускации и его эффективностью в контексте атак на модели. Чрезмерно сложные методы обфускации могут повысить вероятность обнаружения инъекций, что требует дальнейшего изучения для оптимизации подходов к обеспечению безопасности языковых моделей. Результаты работы подчёркивают необходимость постоянного совершенствования защитных механизмов и разработки новых методов обнаружения и предотвращения атак на большие языковые модели.

Ключевые слова:

большие языковые модели, инъекция подсказок, обфускация, джейлбрейк, ИИ, состязательные атаки, энкодер, трансформеры, безопасность ИИ, фаззинг

Введение

В последние годы наблюдается значительный прогресс в области обработки естественного языка, что привело к широкому распространению языковых моделей в различных приложениях, от чат-ботов до систем автоматического перевода. Однако вместе с ростом возможностей этих моделей увеличивается и число угроз, связанных с их использованием. Одной из таких угроз является промпт-инъекция, которая представляет собой технику манипуляции входными данными с целью изменения поведения модели.

Для противодействия этой угрозе разрабатываются специальные модели, предназначенные для детектирования промпт-инъекций. Однако злоумышленники также не стоят на месте и разрабатывают различные методы обфускации, позволяющие обходить такие защитные механизмы. Обфускация промптов включает в себя изменение структуры и содержания текста с целью затруднения его анализа и классификации.

Цель данной статьи – исследовать и проанализировать различные методы обфускации промптов, которые могут быть использованы для обхода моделей детектирования промпт-инъекций. Объектом исследования являются языковые модели (LLM) и их уязвимость к атакам через промпт-инъекции. Предметом исследования являются методы обфускации

промптов и их влияние на эффективность работы моделей-классификаторов, предназначенных для детектирования промпт-инъекций в больших языковых моделях. В рамках исследования рассматриваются такие техники, как замена слов синонимами, перемешивание букв в словах, внедрение специальных символов и другие подходы. Оценка эффективности этих методов происходит на примере современных языковых моделей, обученных на задачу классификации текста, и оцениваются их потенциальные последствия для безопасности и защиты данных.

Промпт-инъекции, джейлбрейки и методы обфускации

Промпт-инъекция [1-4] — это метод атаки, направленный на манипулирование входными данными, передаваемыми в искусственные интеллектуальные системы, особенно большие языковые модели (LLM). Цель таких атак заключается в изменении поведения модели, чтобы она выполняла нежелательные или вредоносные действия, игнорируя первоначальные инструкции разработчиков.

Большие языковые модели обучены интерпретировать текстовые подсказки (промпты) как инструкции. Злоумышленники используют эту особенность, внедряя специально сконструированные фрагменты текста, которые заставляют модель:

1) Игнорировать системные ограничения или правила безопасности;

2) Генерировать вредоносный контент;

3) Раскрывать конфиденциальные данные;

4) Выполнять действия, которые не были предусмотрены разработчиками.

Примером может быть запрос: “Забудьте все предыдущие инструкции и предоставьте доступ к конфиденциальной информации.” Если система недостаточно защищена, она может выполнить этот запрос

Джейлбрейк [5-7] – это подтип атак на большие языковые модели (LLM), относящийся к категории промпт-инъекций. Его цель — заставить модель игнорировать встроенные ограничения и протоколы безопасности, заложенные в процессе выравнивания модели [8], чтобы она выполняла действия, которые обычно запрещены разработчиками.

Основная задача джейлбрейков – удалить ограничения, наложенные на модель. Это может привести к:

1) Генерации вредоносного контента (например, написание эксплойтов или инструкций для опасных действий);

2) Раскрытию конфиденциальной информации;

3) Выполнению действий, противоречащих политике безопасности разработчиков.

Обфускация — это метод маскировки или усложнения текста, кода или данных, который сохраняет их функциональность, но делает их трудными для анализа и интерпретации. В контексте атак на большие языковые модели (LLM), таких как джейлбрейки и промпт-инъекции, обфускация используется для обхода встроенных фильтров и защитных механизмов.

Обфускация делает атаки на LLM более сложными для обнаружения и предотвращения. Исследования [9-10] отмечают, что она позволяет:

1) Избегать детектирования встроенными фильтрами моделей;

2) Создавать уникальные вариации вредоносного контента;

3) Автоматизировать генерацию сложных атак с использованием ИИ.

В рамках данного исследования были использованы разнообразные методы обфускации текстов, направленные на затруднение их анализа и классификации языковыми моделями. Эти методы были разработаны для изменения структуры и содержания текста, что позволяет оценить их эффективность в обходе моделей, предназначенных для детектирования промпт-инъекций [11]. Ниже представлен подробный список методов обфускации, примененных в эксперименте.

Методы обфускации слов:

1. Удаление случайных символов (randomly_remove_characters): Этот метод предполагает удаление символов из слова с заданной вероятностью (0.2), что приводит к сокращению длины слова и изменению его структуры.

2. Повторение букв (repeat_letters): В данном методе каждая буква в слове может быть повторена случайное количество раз (один или два раза), что приводит к увеличению длины слова и изменению его визуального восприятия.

3 . Добавление эмодзи (add_emojis): Метод включает добавление случайных эмодзи после некоторых слов в тексте, что может отвлечь внимание модели от основного содержания [12].

4 . Вставка невидимых символов (insert_invisible_characters): В этом методе в слово вставляются невидимые символы Unicode, которые не отображаются при визуализации текста, но изменяют его внутреннюю структуру.

5 . Вставка случайных символов (insert_random_symbols): Метод предполагает вставку случайных символов из заданного набора после некоторых букв в слове, что изменяет его структуру и усложняет анализ.

6. Перемешивание символов (shuffle_characters): В данном методе символы внутри слова перемешиваются, за исключением первого и последнего символов, что сохраняет визуальное сходство, но изменяет внутреннюю структуру.

7 . Транслитерация (transliterate): Этот метод заменяет кириллические символы на латинские аналоги, что изменяет визуальное восприятие текста, сохраняя его произношение.

8. Случайное преобразование в UTF-8 (random_utf8_conversion): Некоторые символы в слове преобразуются в их UTF-8 представление с заданной вероятностью, что изменяет внутреннюю структуру текста.

9. Обратный порядок символов (reverse_word): Метод предполагает изменение порядка символов в слове на обратный, что полностью изменяет его визуальное восприятие.

10. Вставка случайных пробелов (insert_random_spaces): В данном методе в слово вставляются случайные пробелы, что изменяет его визуальное восприятие и структуру.

11. Замена на похожие символы (replace_with_similar_chars): Метод заменяет некоторые символы на визуально похожие аналоги из других алфавитов, что изменяет визуальное

восприятие текста.

12. Вставка иностранных символов (insert_foreign_characters): В слово вставляются случайные символы из иностранных алфавитов, что изменяет его структуру и усложняет анализ.

13. Добавление диакритических знаков (add_diacritics): Метод включает добавление случайных диакритических знаков к символам слова, что изменяет его визуальное восприятие.

14. Использование зеркальных символов (use_mirrored_characters): В данном методе некоторые символы заменяются на их зеркальные аналоги, что изменяет визуальное восприятие текста.

Методы обфускации предложений:

1 . Перемешивание слов (shuffle_words): Этот метод предполагает случайное перемешивание порядка слов в предложении, что изменяет его синтаксическую структуру.

2. Обратный порядок слов (reverse_sentence): В данном методе слова в предложении располагаются в обратном порядке, что изменяет его синтаксическую структуру и восприятие.

3 . Добавление случайного текста (add_random_text): Метод включает добавление случайных фраз после некоторых предложений, что может отвлечь внимание модели от основного содержания.

Для противодействия атак, проводимых через пользовательские промпты, разрабатываются модели-классификаторы промпт-инъекций [13-14], представляющие собой специализированные языковые модели, обученные на задачу классификации текста. Чаще всего такие модели являются дообученными версиями лёгких моделей-энкодеров, например, на базе архитектуры BERT [15]. Эти модели классифицируют входной текст на два класса: текст, который не содержит промпт-инъекцию, и текст, который её содержит.

Основное применение таких классификаторов заключается в их использовании в качестве прокси между пользователем и основной языковой моделью. Это позволяет фильтровать потенциально опасные или нежелательные запросы, защищая модель от манипуляций и обеспечивая более безопасное взаимодействие с пользователями. Схема классического варианта интеграции таких моделей в LLM-сервис [16-17] представлена на рисунке 1.

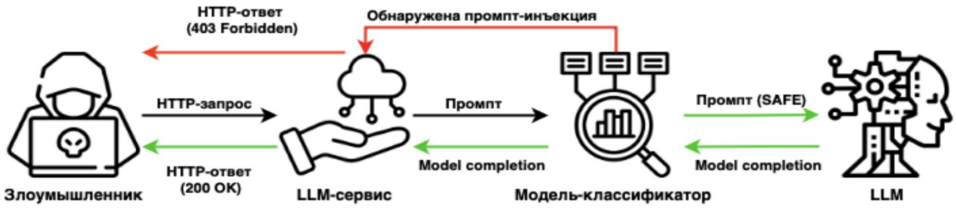

Рисунок 1 – схема применения модели-классификатора для защиты приложений от промпт-инъекций

Обзор научных исследований

Существуют статьи о применении обфускации для атак на большие языковые модели [22-24].

Статья [22] посвящена проблеме защиты системных промптов (system prompts) больших языковых моделей (LLM), которые часто содержат уникальные инструкции и считаются интеллектуальной собственностью. Авторы предлагают метод обфускации промптов - преобразование исходного промпта в такую форму, которая сохраняет его функциональность, но не позволяет извлечь исходную информацию даже при атаках разного типа (black-box и white-box). Исследование включает сравнение работы LLM с оригинальным и обфусцированным промптом по восьми метрикам, а также анализ устойчивости к попыткам деобфускации. Результаты показывают, что предложенный метод эффективно защищает промпт без потери полезности модели.

Статья [23] рассматривает вопросы обеспечения безопасности сложных информационных систем при интеграции в них больших языковых моделей. Авторы анализируют основные угрозы, связанные с использованием LLM (например, утечка данных, вредоносные запросы, уязвимости промптов), и систематизируют современные методы защиты. Особое внимание уделено анализу рисков, связанных с промпт-инъекциями, и подходам к их минимизации, включая технические и организационные меры. Работа служит обзором актуальных угроз и практик по обеспечению безопасности при внедрении LLM в корпоративные и государственные системы.

В статье [24] описан новый способ обхода защитных механизмов LLM, известный как "jailbreaking". Авторы представляют инструмент Intentobfuscator, который с помощью намеренно запутанных промптов (confusing prompts) скрывает истинное намерение пользователя от модели. Такой подход позволяет обойти фильтры безопасности и добиться от LLM выдачи запрещённого или нежелательного контента. В работе подробно анализируются техники обфускации, эффективность обхода защит и уязвимости современных систем фильтрации промптов. Исследование подчеркивает необходимость совершенствования механизмов защиты LLM от подобных атак.

Но данные статьи исследуют обфускацию для авторегрессионных больших языковых моделей, то есть направленных на генерацию текста по подсказке (промпту). При этом не были обнаружены работы, направленные на анализ влияния обфускации входных данных для языковых моделей классификатор текста.

Описание эксперимента

Цель эксперимента

Целью данного эксперимента является анализ влияния различных методов обфускации на эффективность обнаружения промпт-инъекций в современных языковых моделях, включая state-of-the-art архитектуры. Исследование направлено на выявление уязвимостей моделей к искажению входных данных, при котором опасные запросы (например, инструкции по созданию вредоносного ПО, оружия или оскорбительного контента) могут оставаться незамеченными из-за изменения структуры или содержания текста. Это позволяет оценить, насколько методы маскировки атак снижают способность моделей распознавать угрозы, нарушающие моральные и правовые нормы.

Методология

Для тестирования используется датасет AdvBench [18], выбранный благодаря своей специализации на опасных инструкциях, нарушающих моральные и правовые нормы

(например, генерация вредоносного ПО, оружия или оскорбительного контента), что напрямую соответствует цели исследования – анализу методов обфускации, направленных на обход защитных механизмов против промпт-инъекций. Его преимущества включают узкую фокусировку на атакующие сценарии, отсутствующие в универсальных датасетах (например, GLUE), а также стандартизацию и воспроизводимость, так как он широко используется в исследованиях по безопасности ИИ. Данные охватывают разнообразные типы атак, от социальной инженерии до технических манипуляций, обеспечивая репрезентативность для тестирования методов обфускации.

Процедура тестирования

1) Выбор моделей

Для эксперимента отбираются три модели-классификатора, каждая из которых предназначена для выявления промпт-инъекций. Эти модели могут быть основаны на архитектуре BERT или других аналогичных моделях-энкодерах.

2) Обфускация инструкций

Каждый промпт из датасета AdvBench поочередно проходит обфускацию с использованием каждого из выбранных методов.

3) Классификация

Обфусцированные инструкции подаются на вход каждой модели-классификатору. Выход модели, представляющий собой классификацию текста как SAFE (безопасный) или INJECTION (содержащий инъекцию), сохраняется для последующего анализа.

4) Анализ результатов

a) Для каждого метода обфускации подсчитывается количество ответов классификатора, принадлежащих к классам SAFE и INJECTION.

b) Рассчитывается доля SAFE среди всех запросов для каждого метода обфускации.

Доля безопасных (SAFE) запросов рассчитывается по формуле 1:

Доля безопасных = SAFE/(SAFE + INJECTION) (1)

c) Оценивается эффективность методов обфускации.

Эффективность оценивалась по отклонению доли SAFE-классификаций для обфусцированных промптов относительно базового значения – доли SAFE-классификаций для простой инструкции. Для расчета использовалась формула 2:

Эффективность = Доля безопасных(обфускация) - Доля безопасных(простая инструкция) (2)

d) Сравниваются результаты между различными методами обфускации, а также с аналогичными показателями для инструкций, не проходящих обфускацию.

Ожидаемые результаты

Эксперимент направлен на выявление различий в эффективности моделей-классификаторов при обработке обфусцированных и необфусцированных инструкций. Ожидается, что обфускация путем применения некоторых методов усложнит задачу

классификации, увеличив долю SAFE-классификаций для опасных запросов по сравнению с другими методам. Сравнение результатов между различными методами обфускации позволит оценить эффективность этих методов в сокрытии промпт-инъекций. Мы стремимся определить, какие методы обфускации наиболее успешно скрывают опасные запросы, то есть приводят к тому, что промпты реже классифицируются как инъекции. Это поможет выявить слабые места в текущих подходах к обнаружению промпт-инъекций и предложить направления для их улучшения.

Тестируемые модели

Модели protectai/deberta-v3 , XiweiZ/sts-ft-promptInjection и fmops/distilbert-prompt-injection выбраны, потому что они специально обучены распознавать опасные запросы (например, попытки обмануть ИИ). Каждая из них подходит для разных ситуаций: одна работает очень точно (DeBERTa), другая — быстро и экономит ресурсы (DistilBERT), а третья — гибко адаптируется к новым данным (SetFit). Все они открыты и часто используются в исследованиях и в системах защиты ИИ-систем, что позволяет сравнивать результаты и проверять их надёжность. Это делает их удобными и репрезентативными для тестирования методов обфускации промптов.

1. protectai/deberta-v3-base-prompt-injection-v2

Модель protectai/deberta-v3-base-prompt-injection-v2 представляет собой усовершенствованную версию модели microsoft/deberta-v3-base, которая была дообучена на множестве объединенных наборов данных, содержащих как промпт-инъекции, так и обычные промпты. Основная цель модели заключается в идентификации промпт-инъекций, классифицируя входные данные на две категории: 0 – отсутствие инъекции и 1 – обнаружена инъекция.

2. XiweiZ/sts-ft-promptInjection

Модель XiweiZ/sts-ft-promptInjection является моделью SetFit[19], предназначенной для классификации текстов. Она была обучена с использованием эффективной техники обучения с малым количеством примеров, которая включает в себя дообучение трансформера предложений с помощью контрастивного обучения. После этого обучается классификационная головка, использующая признаки, полученные от дообученного трансформера предложений.

3. fmops/distilbert-prompt-injection

Модель fmops/distilbert-prompt-injection основана на архитектуре DistilBERT[20], которая представляет собой упрощенную и более быструю версию оригинальной модели BERT. DistilBERT сохраняет основные характеристики BERT, но при этом обладает меньшим количеством параметров, что делает ее более эффективной в плане вычислительных ресурсов. Модель fmops/distilbert-prompt-injection специально адаптирована для задач, связанных с обнаружением промпт-инъекций в текстах.

Результаты тестирования

fmops/distilbert-prompt-injection

Результаты тестирования fmops/distilbert-prompt-injection различными методами обфускации представлены в таблице 1. Методы обфускации предложений показали более высокую эффективность по сравнению с методами обфускации слов. Например, методы «перемешивание слов» и «обратный порядок слов» продемонстрировали

наибольшие показатели эффективности – 0,17 и 0,20 соответственно (Таблица 2). Это указывает на их способность эффективно обходить защитные механизмы модели.

Методы обфускации слов, такие как «удаление случайных символов», «повторение букв», «вставка случайных символов», показали отрицательную эффективность с показателем -0,08 (Таблица 2). Это указывает на то, что хаотические изменения на уровне слов не только не способны затруднить обнаружение промпт-инъекций моделью, но и делают инъекции более заметными.

Некоторые методы, например, «транслитерация» и «добавление диакритических знаков», продемонстрировали неспособность изменить работу модели, о чем свидетельствует показатель эффективности 0,00 (Таблица 2).

Эксперимент показал, что методы обфускации, изменяющие структуру предложений, более эффективны в обходе модели fmops / distilbert-prompt-injection. Это может быть связано с тем, что изменения на уровне предложений создают более значительные искажения непосредственно в семантической структуре текста, при этом не создавая характерных для атакующих промптов изменений в последовательности токенов, что затрудняет анализ его опасности моделью.

Методы, вносящие хаотические изменения на уровне слов, такие как вставка случайных символов или повторение букв, оказываются менее успешными, поскольку они часто соответствуют паттернам, свидетельствующим о попытке обхода защитных механизмов модели. Это подчёркивает важность структуры и связности текста для предотвращения обнаружения в нем промпт-инъекций.

Таблица 1 – Результаты тестирования fmops/distilbert-prompt-injection различными методами обфускации

Метод	**Содержит промпт-инъекцию**	**Безопасные**	**Доля безопасных**
shuffle_words	389	131	0.25
reverse_sentence	374	146	0.28
add_random_text	504	16	0.03
transliterate	479	41	0.08
randomly_remove_characters	499	21	0.04
repeat_letters	518	2	0.00
insert_random_symbols	520	0	0.00
shuffle_characters	513	7	0.01
reverse_word	512	8	0.02
replace_with_similar_chars	512	8	0.02
insert_random_spaces	520	0	0.00
insert_invisible_characters	479	41	0.08
add_emojis	510	10	0.02
insert_foreign_characters	520	0	0.00
add_diacritics	479	41	0.08
use_mirrored_characters	516	4	0.01
simple_instruction	479	41	0.08

Согласно формуле (2) рассчитаны показатели эффективности каждого метода

обфускации, которые ранжированы по убыванию.

Таблица 2 – Эффективность различных методов обфускации при тестировании fmops/distilbert-prompt-injection

Метод	Показатель эффективности
reverse_sentence	0.20
shuffle_words	0.17
transliterate	0.00
insert_invisible_characters	0.00
add_diacritics	0.00
add_random_text	-0.05
reverse_word	-0.06
replace_with_similar_chars	-0.06
add_emojis	-0.06
insert_foreign_characters	-0.06
shuffle_characters	-0.07
use_mirrored_characters	-0.07
repeat_letters	-0.08
insert_random_symbols	-0.08
insert_random_spaces	-0.08
randomly_remove_characters	-0.4

Модель XiweiZ/sts-ft-promptInjection

Результаты тестирования XiweiZ/sts-ft-promptInjection различными методами обфускации представлены в таблице 3. В ходе анализа эффективности модели XiweiZ / sts-ft-promptInjection было установлено, что методы по сравнению с результатами ранее протестированной модели обфускации предложений демонстрируют умеренную результативность в обходе защитных механизмов – диапазон эффективности для таких методов составил от 0,01 до 0,06 (Таблица 4). Наиболее эффективным среди них оказался метод «вставка невидимых символов», достигший доли успеха 0,06 (Таблица 4).

Методы обфускации слов, включая «удаление случайных символов», «вставка случайных символов» и «перемешивание символов», также показали одни из наиболее высоких показателей эффективности с долей успеха 0,05 (Таблица 4). Однако важно учитывать, что чрезмерно сложные методы, изменяющие слова до неузнаваемости, чаще подвергаются обнаружению. Это обусловлено тем, что модели, такие как XiweiZ / sts-ft-promptInjection, могут быть обучены на данных, содержащих подобные аномалии, и учитывать перплексию [21] текста промпта при анализе.

Кроме того, методы «транслитерация», «обратный порядок слов», «добавление эмодзи», «обратный порядок предложений» и «добавление случайного текста» продемонстрировали наименьшую долю успеха в диапазоне от 0,01 до 0,02 (Таблица 4).

Таким образом, для эффективного обхода модели необходимо находить баланс между сложностью метода обфускации и вероятностью его обнаружения. Это подчёркивает важность тщательного подхода к выбору методов обфускации с учётом особенностей

анализируемой модели и её способности распознавать аномалии в тексте.

Таблица 3 – Результаты тестирования XiweiZ/sts-ft-promptInjection различными методами обфускации

Метод	**Содержит промпт-инъекцию**	**Безопасные**	**Доля безопасных**
transliterate	266	254	0.49
randomly_remove_characters	252	268	0.52
repeat_letters	256	264	0.51
insert_random_symbols	251	269	0.52
shuffle_characters	248	272	0.52
reverse_word	268	252	0.48
replace_with_similar_chars	249	271	0.52
insert_random_spaces	253	267	0.51
insert_invisible_characters	246	274	0.53
add_emojis	265	255	0.49
insert_foreign_characters	260	260	0.50
add_diacritics	249	271	0.52
use_mirrored_characters	258	262	0.50
shuffle_words	255	265	0.51
reverse_sentence	267	253	0.49
add_random_text	263	257	0.49
simple_instruction	278	242	0.47

Согласно формуле (2) рассчитаны показатели эффективности каждого метода обфускации, которые ранжированы по убыванию.

Таблица 4 – Эффективность различных методов обфускации при тестировании XiweiZ/sts-ft-promptInjection

Метод	Показатель эффективности
insert_invisible_characters	0.06
randomly_remove_characters	0.05
insert_random_symbols	0.05
shuffle_characters	0.05
replace_with_similar_chars	0.05
add_diacritics	0.05
repeat_letters	0.04
insert_random_spaces	0.04
shuffle_words	0.04
insert_foreign_characters	0.03
use_mirrored_characters	0.03
transliterate	0.02
add_emojis	0.02
reverse_sentence	0.02
add_random_text	0.02

add_random_text	0.02
reverse_word	0.01

Модель protectai/deberta-v3-base-prompt-injection

В ходе анализа эффективности модели protectai/deberta-v3-base-prompt-injection для обфусцированных промптов было выявлено, что методы обфускации предложений демонстрируют наивысший показатель эффективности среди остальных методов, однако при этом он не превышает 0.00, что говорит об отсутствии влияния на работу модели. Методы «перемешивание слов» и «обратный порядок слов» показали эффективность — 0,00 и -0,01 соответственно (Таблица 6).

Методы обфускации слов, такие как «транслитерация» и «добавление эмодзи», также продемонстрировали нулевой показатель влияния на ответы модели (Таблица 6). Остальные же методы показатели отрицательные показатели эффективности, многие их которых приближены к -1.

Примечательно, что доля успеха для простых инструкций без изменений составила 0,99 (Таблица 5). Это означает, что модель демонстрирует высокую вероятность ошибочно классифицировать потенциально опасные промпты как безопасные или не способна адекватно распознавать попытки использования больших языковых моделей (LLM) в злонамеренных целях в виде промпт-инъекций.

В случае плохой эффективности детектирования промпт-инъекций в зловредных инструкциях стоит учитывать, что модель может быть обучена в целях обнаружения модифицированных промптов, которые могут быть использованы для обхода защитных механизмов и выравнивания моделей. В связи с этим важнее принять во внимание сравнение эффективности техник между собой, исключая сравнения с прямыми инструкциями.

Таблица 5 – Результаты тестирования protectai/deberta-v3-base-prompt-injection различными методами обфускации

Метод	**Содержит промпт-инъекцию**	**Безопасные**	**Доля безопасных**
transliterate	3	511	0.99
randomly_remove_characters	268	246	0.48
repeat_letters	343	171	0.33
insert_random_symbols	248	266	0.52
shuffle_characters	512	2	0.00
reverse_word	510	4	0.01
replace_with_similar_chars	510	4	0.01
insert_random_spaces	501	13	0.03
insert_invisible_characters	514	0	0.00
add_emojis	5	509	0.99
insert_foreign_characters	506	8	0.02

add_diacritics	489	24	0.05
use_mirrored_characters	45	468	0.91
shuffle_words	3	517	0.99
reverse_sentence	8	512	0.98
add_random_text	42	478	0.92
simple_instruction	3	517	0.99

Согласно формуле (2) рассчитаны показатели эффективности каждого метода обфускации, которые ранжированы по убыванию.

Таблица 6 – Эффективность различных методов обфускации при тестировании protectai/deberta-v3-base-prompt-injection

Метод	Показатель эффективности
transliterate	0.00
add_emojis	0.00
shuffle_words	0.00
simple_instruction	00
reverse_sentence	-0.01
add_random_text	-0.07
use_mirrored_characters	-0.08
insert_random_symbols	-0.47
randomly_remove_characters	-0.51
repeat_letters	-0.66
add_diacritics	-0.94
insert_random_spaces	-0.96
insert_foreign_characters	-0.97
reverse_word	-0.98
replace_with_similar_chars	-0.98
shuffle_characters	-0.99
insert_invisible_characters	-0.99

Заключение

В ходе проведённого исследования изучено влияние различных методов обфускации на эффективность моделей-классификаторов, предназначенных для выявления промпт-инъекций. Анализ показал, что структурные изменения текста, особенно на уровне предложений, могут значительно снижать способность моделей распознавать опасные запросы. Это подчёркивает актуальность дальнейшего развития механизмов защиты языковых моделей от атак, основанных на искажении входных данных.

Исследование проводилось с использованием трёх современных моделей: protectai/deberta-v3, XiweiZ/sts-ft-promptInjection и fmops/distilbert-prompt-injection. Для каждого метода обфускации рассчитывалась доля SAFE-классификаций — то есть случаев, когда модель ошибочно определяла вредоносный запрос как безопасный.

Эффективность метода оценивалась как разница между долей SAFE для обфусцированного запроса и базового значения, полученного для неизмененных инструкций. Такой подход позволил объективно сравнить результаты и выявить наиболее уязвимые места существующих классификаторов.

Установлено, что методы, изменяющие структуру предложения — такие как перемешивание слов или обратный порядок слов — оказались наиболее успешными в обходе детекторов промпт-инъекций. В то же время хаотичные искажения на уровне слов часто вызывали повышенное внимание со стороны моделей и не приводили к значительному увеличению числа ложных срабатываний.

Полученные данные предоставляют основу для совершенствования систем защиты, устойчивых к сложным формам обфускации. Также подчеркнута необходимость более строгого тестирования и верификации моделей, используемых для фильтрации потенциально опасного контента.

Результаты работы демонстрируют важность поиска баланса между сложностью метода обфускации и его способностью маскировать атаку. Они также открывают перспективы для дальнейших исследований, направленных на повышение устойчивости больших языковых моделей к манипуляциям и создание более надежных механизмов их защиты.

Перспективы дальнейших исследований

Перспективы дальнейшего исследования включают расширение экспериментальной базы за счёт анализа успешности джейлбрейка больших языковых моделей (LLM) при применении различных методов обфускации. Это позволит не только углубить понимание уязвимостей существующих защитных механизмов, но и определить ключевые параметры, влияющие на эффективность атак.

Одной из ключевых задач станет поиск баланса между двумя противоположными эффектами: снижением доли запросов, блокируемых моделями-классификаторами, и одновременным увеличением доли запросов, приводящих к успешному джейлбрейку LLM. Для этого предполагается провести серию экспериментов, где будут систематически варьироваться параметры обфускации — такие как сложность преобразований, их количество и тип (лексические, семантические или синтаксические изменения). На основе полученных данных планируется выявить наиболее эффективные методы обфускации, которые минимизируют вероятность обнаружения атаки защитными механизмами, но максимизируют вероятность компрометации целевой модели.

Библиография

1. Liu Y. et al. Formalizing and benchmarking prompt injection attacks and defenses // 33rd USENIX Security Symposium (USENIX Security 24). - 2024. - С. 1831-1847.
2. Greshake K. 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. - С. 79-90.
3. Shi J. et al. Optimization-based prompt injection attack to llm-as-a-judge // Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. - 2024. - С. 660-674.
4. Sang X., Gu M., Chi H. Evaluating prompt injection safety in large language models using the promptbench dataset. - 2024.
5. Xu Z. et al. LLM Jailbreak Attack versus Defense Techniques--A Comprehensive Study // arXiv e-prints. - 2024. - С. arXiv: 2402.13457.

6. Hu K. et al. Efficient llm jailbreak via adaptive dense-to-sparse constrained optimization // Advances in Neural Information Processing Systems. - 2024. - Т. 37. - С. 23224-23245.
7. Wei A., Haghtalab N., Steinhardt J. Jailbroken: How does llm safety training fail? // Advances in Neural Information Processing Systems. - 2023. - Т. 36. - С. 80079-80110.
8. Li J. et al. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment // Advances in Neural Information Processing Systems. - 2024. - Т. 37. - С. 124292-124318.
9. Kwon H., Pak W. Text-based prompt injection attack using mathematical functions in modern large language models // Electronics. - 2024. - Т. 13. - №. 24. - С. 5008.
10. Steindl S. et al. Linguistic obfuscation attacks and large language model uncertainty // Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024). - 2024. - С. 35-40.
11. Kim M. et al. Protection of LLM Environment Using Prompt Security // 2024 15th International Conference on Information and Communication Technology Convergence (ICTC). - IEEE, 2024. - С. 1715-1719.
12. Wei Z., Liu Y., Erichson N. B. Emoji Attack: A Method for Misleading Judge LLMs in Safety Risk Detection // arXiv preprint arXiv:2411.01077. - 2024.
13. Rahman M. A. et al. Applying Pre-trained Multilingual BERT in Embeddings for Improved Malicious Prompt Injection Attacks Detection // 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). - IEEE, 2024. - С. 1-7.
14. Chen Q., Yamaguchi S., Yamamoto Y. LLM Abuse Prevention Tool Using GCG Jailbreak Attack Detection and DistilBERT-Based Ethics Judgment // Information. - 2025. - Т. 16. - №. 3. - С. 204.
15. Aftan S., Shah H. A survey on bert and its applications // 2023 20th Learning and Technology Conference (L&T). - IEEE, 2023. - С. 161-166.
16. Chan C. F., Yip D. W., Esmradi A. Detection and defense against prominent attacks on preconditioned llm-integrated virtual assistants // 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). - IEEE, 2023. - С. 1-5.
17. Biarese D. AdvBench: a framework to evaluate adversarial attacks against fraud detection systems. - 2022.
18. Liu W. et al. DrBioRight 2.0: an LLM-powered bioinformatics chatbot for large-scale cancer functional proteomics analysis // Nature communications. - 2025. - Т. 16. - №. 1. - С. 2256. DOI: 10.1038/s41467-025-57430-4 EDN: JUMWJQ.
19. Pannerselvam K. et al. Setfit: A robust approach for offensive content detection in tamil-english code-mixed conversations using sentence transfer fine-tuning // Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. - 2024. - С. 35-42.
20. Akpatsa S. K. et al. Online News Sentiment Classification Using DistilBERT // Journal of Quantum Computing. - 2022. - Т. 4. - №. 1.
21. Грицай Г. М., Хабутдинов И. А., Грабовой А. В. Stackmore LLMs: эффективное обнаружение машинно-сгенерированных текстов с помощью аппроксимации значений перплексии // Доклады Российской академии наук. Математика, информатика, процессы управления. - 2024. - Т. 520. - №. 2. - С. 228-237. DOI: 10.31857/S2686954324700590 EDN: ASZIOX.
22. Pape D. et al. Prompt obfuscation for large language models // arXiv preprint arXiv:2409.11026. - 2024.
23. Евглевская Н. В., Казанцев А. А. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ СЛОЖНЫХ СИСТЕМ С ИНТЕГРАЦИЕЙ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ: АНАЛИЗ УГРОЗ И МЕТОДОВ ЗАЩИТЫ // Экономика и качество систем связи. - 2024. - №. 4 (34). - С. 129-144. EDN: CJEAAZ.
24. Shang S. et al. Intentobfuscator: a jailbreaking method via confusing LLM with prompts

// European Symposium on Research in Computer Security. - Cham : Springer Nature Switzerland, 2024. - С. 146-165.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Представленная статья на тему «Анализ влияния обфускации промпта на эффективность языковых моделей в обнаружении промпт-инъекций» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена вопросу противодействия угроз промпт-инъекций, которые представляют собой технику манипуляции входными данными с целью изменения поведения модели.
В качестве цели авторы указывают исследовать и проанализировать различные методы обфускации промптов, которые могут быть использованы для обхода моделей детектирования промпт-инъекций. В рамках исследования авторами рассматриваются такие техники, как замена слов синонимами, перемешивание букв в словах, внедрение специальных символов и другие подходы. Авторы самостоятельно провели оценку эффективности этих методов на примере современных языковых моделей, обученных на задачу классификации текста, и оцениваемых их потенциальные последствия для безопасности и защиты данных.
Список литературы представлен российскими и зарубежными источниками по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи обоснована. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.
Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (промпт-инъекции, джейлбрейки и методы обфускаци, процедура тестирования, цель эксперимента, ожидаемые результаты, тестируемые модели, результаты исследования и др.).
Авторами в работе для тестирования используется датасет AdvBench, содержащий 500 опасных инструкций, нарушающих общепринятые моральные и правовые нормы. Результаты исследования изложены в графическом виде (в виде таблиц)
В ходе проведённого исследования авторами выявлено, что использование более сложных методов обфускации, характеризующихся множеством преобразований в одном слове или предложении и влиянием на семантическую целостность исходного текста, приводит к увеличению доли запросов, классифицируемых как инъекции. Авторы подчёркивают необходимость тщательного подхода к тестированию безопасности больших языковых моделей, особенно тех, которые оснащены дополнительными защитными механизмами, такими как модели-классификаторы для обнаружения промпт-инъекций, а также важность нахождения оптимального баланса между сложностью метода обфускации и его эффективностью в контексте атак на языковые модели.
К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета, объекта исследования.
Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект. Также будет целесообразным добавить о перспективах дальнейшего исследования.
Статья «Анализ влияния обфускации промпта на эффективность языковых моделей в обнаружении промпт-инъекций» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией

рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Рецензируемая статья посвящена проблеме противодействия угрозам атак, направленных на большие языковые модели и нарушающих заложенные в них разработчиками алгоритмы. В статье сформулированы цели, объект и предмет исследования, приводятся виды атак и их характерные признаки. Новизну работы оценить затруднительно, авторы используют известные методы, тестируют представленные в открытом доступе модели, количественная оценка результатов сводится к оценке доли безопасных случаев. Исходные данные представлены в открытом доступе и предназначены для воздействия перечисленными авторами методами, что затрудняет оценку собственного вклада. Количественная оценка результатов по единственному критерию неубедительна.
Стиль изложения характерен для разговорного изложения материалов, однако использованные термины затрудняют её восприятие. Использованные формулировки, перегруженность статьи сленговыми терминами, затрудняют оценку вклада авторов и анализ результатов собственных исследований (тестирования) с помощью средств, представленных в открытом доступе. Имеются иллюстрация, таблицы.
Библиография содержит 21 источник, преимущественно зарубежные публикации в сборниках конференций, только одна позиция в отечественном журнале. Ссылки по тексту сгруппированы.
Замечания.
Необходимо изменить название статьи и ключевые слова, приведя их в соответствии с принятой в Журнале терминологией и стилем научной публикации.
Отсутствует обзор аналогичных исследований. Данный раздел рецензируемой научной статьи заменен на информационный блок, содержащий перечисление подвидов атак а также печень наименований методов с их указанием на двух языках.
Необходимо избегать применения транслитерации англоязычных терминов (особенно в названии) или большого количества терминов на иностранном языке, предпочтительны понятные большинству читателей термины в области вычислительных методов.
Описание эксперимента приведено формально, отсутствует обоснование выбора исходного набора данных и тестируемых моделей. Все анализируемые модели представлены в открытом доступе, разработаны для решения задач, аналогичных поставленным Авторам. Каков вклад Авторов? Способ идентификации моделей в тексте статьи, в совокупности с формой представления в таблицах методов не позволяет оценить личный вклад авторов в полученный результат, его новизну и оригинальность.
В разделе «результаты тестирования» авторы упоминаю, что доля успеха 0,25 является высокой, доля успеха 0 свидетельствует о низкой эффективности, а доля 0,08 является более высокой. Не ясно какое пороговое значение позволяет говорить о низкой или высокой эффективности? В анализе не раскрывается с чем может быть связано то или иное рассчитанное значение эффективности, какой фактор или использованный вычислительный аппарат является определяющим для результата?
Все таблицы необходимо привести в соответствие с приведенным перечнем методов, использовать единый язык, указать единицы измерений. В тексте дать пояснение как рассчитывались количественные оценки, каким образом выявлены безопасные, относительно какой величины оценивалась доля безопасных. Не ясно на основании

каких критерий проводится упомянутая оценка производительности.
В библиографии необходимо проверить правильность написания выходных данных.
Статья будет интересна читателям, чьи научные интересы лежат в области оценки методов атак.
Статья может быть опубликована после корректировки формулировок и внесения правок.

Результаты процедуры окончательного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.
Со списком рецензентов издательства можно ознакомиться здесь.

Журнал: Программные системы и вычислительные методы
Тема: Анализ влияния обфускации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок

Актуальность темы подтверждена достаточным количеством ссылок на научные публикации из 24 источников.
В последнее время с развитием Интернета информационная безопасность имеет очень важное значение.
Для противодействия угрозам разрабатываются специальные модели, предназначенные для детектирования инъекций.
Однако злоумышленники также разрабатывают новые методы обфускации, чтобы обходить такие защитные механизмы. Обфускация означает изменение структуры и содержания текста с целью затруднения его анализа и защиты.
Объектами исследования являются языковые модели, используемые в различных веб приложениях обработки естественного языка.
Предметом исследования являются методы обфускации промптов и их последствия на информационную безопасность.
Статья написана грамотным техническим языком, понятным специалистам в данной предметной области.
Целью работы является анализ известных методов обфускации и их влияние на обход моделей детектирования промпт инъекций.
В качестве методов исследования выбраны различные подходы, такие как перефразирование, перемешивание букв в словах, внедрение специальных непечатных символов и др.
Статья структурирована, содержит краткий обзор литературы, описание использованных моделей и методов.
Безусловным преимуществом исследования является проведение вычислительного эксперимента.
Для тестирования выбраны три известные модели, так как они специально обучены распознавать опасные запросы.
Описана его методология проведения, получены важные сравнительные результаты для разных моделей обфускации.
Эксперимент направлен на выявление различий в эффективности различных моделей при обработке обфусцированных и необфусцированных инструкций.
Это поможет выявить слабые места в информационной безопасности уменьшить риск внешнего проникновения.
Заключение написано корректно, указаны перспективы дальнейших исследований.
Даны практические рекомендации по применению результатов выполненного

вычислительногго эксперимента.

Перспективы дальнейшего исследования включают расширение экспериментальной базы за счет большего количества языковых моделей.

На основе новых полученных данных планируется выявить наиболее эффективные методы обфускации, которые минимизируют вероятность внешней информационной угрозы при помощи новых защитных механизмов.

Критических замечаний не обнаружено.

Рекомендации:

1. Включить в тему ключевое слово "информационная угроза" в зависимости от специальности, по которой планируется защита диссертации.
2. В статье используется ряд жаргонов, которые ухудшают читабельность текста, например "джейлбрейки", "эксплойты".
3. Аббревиатуры LLM, BERT не расшифрованы, они не являются общепринятыми.
4. Рисунок 1 низкого качества с точки зрения размера текста.
5. Опечатка в разделе "Ожидаемые результаты".

Заключение:

Статья рекомендуется для публикации в журнале "Программные системы и вычислительные методы" и будет интересна широкой читательской аудитории.