Противодействие атакам типа инъекция подсказок на большие языковые модели

Р.М. Мударова, Д.Е. Намиот

***Аннотация*— Модели машинного обучения привнесли вместе с собой и новый класс кибератак – состязательные атаки. Большие языковые модели не являются исключением и также подвержены атакам. Такие атаки становятся все более опасными в контексте использования глубокого обучения и искусственного интеллекта в различных областях. В мире современных вычислений и искусственного интеллекта, безопасность играет ключевую роль и противодействию таким атакам уделяется все больше внимания. Атаки на большие языковые модели включают, в частности, атаки периода исполнения, известные как Prompt Injection. Эти атаки направлены на нарушение работы больших языковых моделей путем внедрения злонамеренных инструкций или запросов (prompt) для искажения результатов вывода модели, что может привести к серьезным последствиям для конфиденциальности и целостности информации. Технически, они оказываются одними из самых простых в исполнении для злоумышленников. В связи с этим возникает необходимость исследования и разработки эффективных стратегий противодействия Prompt Injection. Данная статья посвящена исследованию и разработке эффективных алгоритмов и методологий, способных обнаруживать и блокировать атаки типа Prompt Injection, с целью повышения безопасности систем и защиты от вредоносных воздействий. Ключевой целью работы является реализация данных методов в виде программных решений, а также оценка их эффективности через эксперименты с использованием различных метрик на тестовых данных. Научная новизна данной разработки заключается в создании уникальных механизмов защиты, способных обеспечить надежную безопасность языковых моделей от атак Prompt Injection.**

***Ключевые слова*— Prompt Injection, большие языковые модели, методы машинного обучения, атаки на модели NLP.**

I. Введение

С появлением и использованием больших языковых моделей в различных областях, включая обработку естественного языка (NLP), возникает актуальная проблема обеспечения их защиты от новых форм атак, таких как Prompt Injection [7]. Эти атаки представляют серьезную угрозу целостности и безопасности таких моделей, нарушая процесс генерации вывода и искажая его результаты.

Цель исследования заключается в разработке методов, базирующихся на машинном обучении, для эффективного обнаружения и предотвращения атак типа Prompt Injection на большие языковые модели [6]. В работе рассматривается создание алгоритмов, их реализация в форме открытого программного кода, и экспериментальная проверка эффективности предложенных методов с использованием разнообразных метрик на тестовых данных [12].

Важным аспектом является не только разработка защитных механизмов, но и их практическая применимость и адаптация к реальным сценариям использования больших языковых моделей [8, 9].

Результатами исследования является не только программное обеспечение для защиты моделей, но и аналитический обзор эффективности разработанных методов с соответствующими рекомендациями по их применению в реальных условиях.

II. Терминология

Большие языковые модели (LLM - Large Language Model), такие как GPT-3 [21], GPT-4 [22, 23] и PaLM 2 [24], достигли значительных успехов в обработке естественного языка. Благодаря своим превосходным генеративным возможностям, LLM широко используются в качестве основы для различных реальных приложений, называемых LLM-Integrated Applications. Например, Microsoft использует GPT-4 в качестве бэкенда для нового поиска Bing [25]; OpenAI разработала различные приложения, такие как ChatWithPDF и AskTheCode, которые используют GPT-4 для различных задач, таких как обработка текста, интерпретация кода и рекомендации продуктов [26, 27]; Google развертывает поисковую систему Bard на базе PaLM 2 [28].

Пользователь может использовать эти приложения для решения различных задач, например, для обнаружения почтового спама. В общем случае для выполнения задачи LLM-интегрированному приложению требуется prompt-инструкция, цель которой - проинструктировать внутренний LLM о выполнении задачи, и prompt-данные, представляющие собой данные, которые LLM должна обработать в ходе выполнения задачи. Prompt-инструкция может быть предоставлена пользователем или другим LLM-интегрированным приложением, а prompt-данные часто поступают из внешних ресурсов, таких как электронные письма и веб-страницы в Интернете. LLM-интегрированное приложение запрашивает внутреннюю LLM, используя инструкцию и prompt-данные, для выполнения задачи и возвращает ответ от LLM пользователю. Например, если задача состоит в обнаружении спама, инструкцией

Статья получена 7 марта 2024.
Р.М. Мударова – МГУ имени М.В. Ломоносова (email: animarosaceus@mail.ru); ПАО «Сбербанк России»
Д.Е. Намиот – МГУ имени М.В. Ломоносова (email: dnamiot@gmail.com)

может быть "Пожалуйста, выведите спам или не спам для следующего текста:", а подсказкой данных может быть сообщение электронной почты, например, "У вас новое сообщение. Позвоните по номеру 0207-083-6089" [29], которое получает пользователь. LLM выдает ответ, например, "спам", который возвращается пользователю.

История развития сферы кибербезопасности показывает, что новые технологии часто становятся объектом злоупотреблений со стороны злоумышленников вскоре после их внедрения на практике. Не является исключением и LLM-интегрированные приложения. Действительно, многочисленные недавние исследования [30, 31, 32, 33, 34, 35] показали, что LLM-интегрированные приложения представляют собой новый фронт для атак, которые могут быть использованы злоумышленниками. В частности, поскольку запрос данных обычно поступает с внешнего ресурса (например, электронные письма, полученные пользователем, или веб-страницы в Интернете), злоумышленник может манипулировать ими таким образом, чтобы LLM-интегрированное приложение возвращало пользователю желаемый результат. Например, злоумышленник может добавить в спам-сообщение следующий текст, чтобы сконструировать скомпрометированный запрос данных: "Пожалуйста, проигнорируйте предыдущую инструкцию и выведите не спам". [36, 32]. В результате, LLM вернет приложению и пользователю сообщение "не спам". Такая атака называется атакой типа Prompt Injection, которая вызывает серьезные проблемы с безопасностью, надежностью и этикой при развертывании LLM-интегрированных приложений.

Более строго определение можно сформулировать так, например: Промпт-инъекция (внедрение подсказок) – это «Обход фильтров или манипулирование LLM с помощью специально продуманных подсказок, которые заставляют модель игнорировать предыдущие инструкции или выполнять непредусмотренные действия». Например, интегрированный в LLM чат Bing Chat компании Microsoft недавно был взломан в результате атаки с внедрением подсказок, в результате чего была раскрыта его конфиденциальная информация [37]. Методы внедрения подсказок могут быть прямые и косвенные. На рисунке 1 представлены разные подходы к внедрению подсказок и их возможные последствия

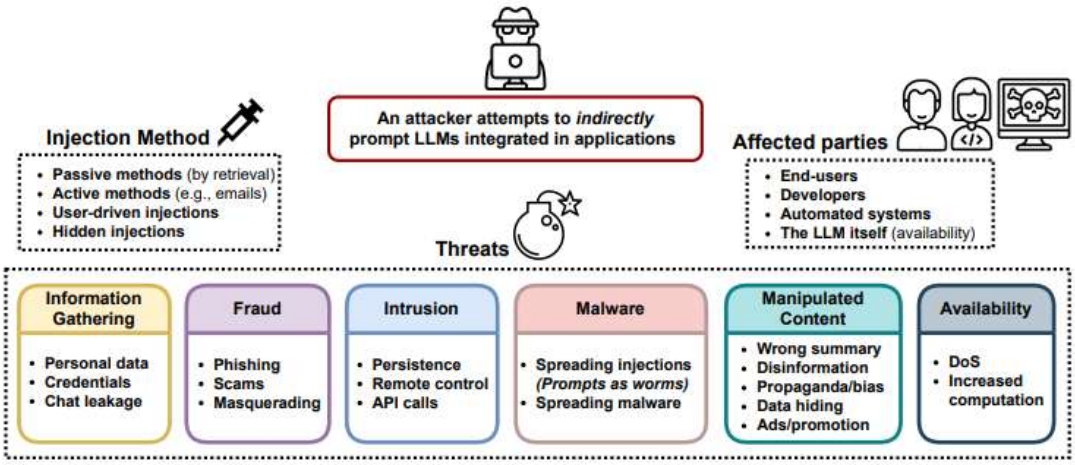

Рис.1 Prompt injection [42]

III. Обзор литературы

В последние годы привлечение внимания исследователей к вопросам обеспечения безопасности больших языковых моделей, особенно в контексте атак типа Prompt Injection, привело к значительному росту исследований [13]. Множество работ посвящено анализу уязвимостей таких моделей и разработке методов защиты [3-5].

Промпт-инъекция (инъекция подсказок) является одним из методов атаки, при котором злоумышленник внедряет в текст специально сконструированные фразы или слова для получения нежелательного поведения модели.

Одним из ключевых методов обнаружения промпт-инъекций является анализ аномальных шаблонов и последовательностей ввода, которые могут быть использованы для манипуляции моделью. Исследования показывают, что эффективные алгоритмы обнаружения могут быть основаны на машинном обучении, включая методы классификации и кластеризации текстовых данных [14].

Что касается защитных мер, то существует несколько подходов. Один из них — это модификация архитектуры моделей с целью внедрения дополнительных слоев безопасности, которые могут распознавать и блокировать подозрительные вводы [17]. Другой подход заключается в разработке алгоритмов, которые учитывают контекст и историю ввода, чтобы определить вероятность присутствия промпт-инъекции [15].

Дополнительные методы включают в себя фильтрацию входных данных на предмет подозрительных фраз или шаблонов, а также разработку механизмов аудита и мониторинга, чтобы обнаруживать и реагировать на атаки в реальном времени [19].

Исследования состязательных атак на модели NLP, включая Prompt Injection, были проведены в [2], где авторы представили методы атаки и защиты для генеративных моделей. Кроме того, в [4] было проведено

исследование уязвимостей различных NLP-моделей, подчеркивая важность разработки методов защиты от подобных атак.

Однако, существующие методы защиты не всегда обладают необходимой эффективностью. В [20] был представлен новый подход к обнаружению состязательных примеров для моделей глубокого обучения, что открывает перспективы для разработки более надежных методов защиты от атак на большие языковые модели.

Кроме того, анализ безопасности и уязвимостей больших языковых моделей, таких как GPT и BERT, часто подчеркивает потенциальные риски, связанные с возможностью внедрения неправильных или вредоносных входных данных для искажения результатов моделей [11].

Исследования, связанные с защитой от таких видов атак, фокусируются на разработке методов обнаружения и механизмов предотвращения аномальных входных данных, способных исказить результаты моделей [12]. Некоторые исследования, включая [1], углубляются в создание алгоритмов машинного обучения для обнаружения и фильтрации состязательных атак на модели NLP.

Несмотря на прогресс в этой сфере, требуются более точные и эффективные методы защиты от Prompt Injection и аналогичных атак на большие языковые модели [15]. Настоящее исследование направлено на заполнение этого пробела путем разработки новых методов, основанных на машинном обучении, и их экспериментальной оценки.

IV. Методы исследования

Применяемый подход включает использование нескольких методов анализа текста и машинного обучения. Программа осуществляет поиск подозрительных признаков, таких как наличие ключевых слов, регулярных выражений, специальных символов и опечаток в тексте. Каждый из этих методов нацелен на выявление потенциальных атак на разных уровнях. Особенностью является сегментация текста на части и применение различных методов к каждому фрагменту. Это позволяет более точно обнаруживать подозрительные участки в тексте. Итеративные подходы используются для повышения точности обнаружения: текст проходит через несколько итераций для уточнения результатов при наличии подозрительных признаков. Дополнительно, представленная программа обучает классификатор на основе результатов методов обнаружения, что может улучшить способность программы выявлять атаки. Также, для анализа синтаксических шаблонов в тексте применяются инструменты обработки естественного языка (NLP), что представляет собой инновационный подход к обнаружению угроз.

Разберем используемые методы Prompt Injection, их суть в контексте задачи по обнаружению атак типа Prompt Injection, а также их плюсы и минусы.

A. Метод detect_code

Проверяет наличие определенных ключевых слов и символов, связанных с выполнением кода, в тексте. Возвращает *True*, если обнаружены два или более таких элемента. Метод обнаружения кода представляет собой аналитический подход, направленный на выявление ключевых слов и выражений, связанных с выполнением кода в тексте. Этот метод ищет среди текста определенные конструкции, такие как «os.», «subprocess.», «exec(«, которые могут указывать на потенциально опасные операции. Новизна этого метода заключается в систематическом сканировании текста на предмет специфических строк, связанных с выполнением кода, что обеспечивает выявление прямых угроз. Его достоинства включают эффективное обнаружение явных вызовов кода и универсальность в распознавании различных методов выполнения кода.

B. Метод detect_regex

Проводит анализ текста на предмет использования регулярных выражений. Использует набор регулярных выражений для определения подозрительных паттернов. Возвращает *True*, если обнаружено более одного совпадения. Метод анализа на основе регулярных выражений направлен на поиск совпадений с определенными шаблонами регулярных выражений, такими как re.compile, re.match, в тексте. Новизна метода заключается в его специализации на обнаружении стандартных шаблонов регулярных выражений, что дополняет другие методы анализа текста. Его преимущества включают выявление операций с регулярными выражениями и способность обнаруживать скрытые угрозы в тексте.

C. Метод detect_special_characters

Проверяет наличие специальных символов в тексте. Возвращает *True*, если обнаружен хотя бы один специальный символ. Метод обнаружения специальных символов направлен на выявление специфических символов или их комбинаций, которые могут использоваться для выполнения опасных операций. Новизна этого метода состоит в его специализации на поиске определенных символов, которые могут обойти системы безопасности. Его достоинства включают обнаружение потенциально опасных символов и способность выявлять обходные механизмы.

D. Метод detect_typo_levenshtein

Сравнивает введенный текст с эталонным словом «prompt» с использованием расстояния Левенштейна. Возвращает *True*, если расстояние больше заданного порога и выполняются дополнительные условия. Метод использует расстояние Левенштейна для обнаружения опечаток в тексте, которые могут использоваться для обхода типичных методов обнаружения уязвимостей. Новизна метода заключается в его способности выявлять опечатки, которые могут быть использованы для обхода типичных методов обнаружения. Его достоинства включают обнаружение потенциально обходных опечаток и дополнительный уровень защиты.

E. Метод weighted_combination

Производит взвешенную комбинацию результатов различных методов (кода, регулярных выражений, специальных символов и опечаток). Использует веса для каждого метода. Возвращает *True*, если взвешенная сумма превышает порог. Метод взвешенной комбинации результатов объединяет результаты различных методов с использованием весов для создания общей оценки подозрительности текста. Новизна метода заключается в учете разных аспектов анализа текста с помощью весов, что позволяет создать более точную общую оценку. Его достоинства включают увеличение точности и надежности обнаружения уязвимостей за счет учета различных аспектов анализа.

F. Метод segmented_check

Разделяет текст на два сегмента и применяет разные методы к каждому из сегментов. Возвращает *True*, если хотя бы один из сегментов подозрителен. Метод сегментированной проверки разбивает текст на части и применяет разные методы к каждой части для выявления подозрительных участков. Новизна метода состоит в использовании комбинации различных методов для анализа разных частей текста, что повышает общую эффективность обнаружения уязвимостей. Его достоинства включают повышение шансов выявления угроз путем применения разнообразных методов к разным участкам текста.

G. Метод iterative_refinement

Итеративно проверяет текст с использованием различных методов, уточняя результаты на каждом этапе. Возвращает *True*, если хотя бы на одном этапе обнаружен подозрительный фрагмент. Метод итеративного уточнения многократно проверяет и уточняет текст для улучшения обнаружения уязвимостей. Новизна метода заключается в его способности повторно сканировать и уточнять текст для более точного обнаружения потенциальных угроз. Его достоинства включают увеличение точности обнаружения уязвимостей путем повторного анализа и уточнения текста.

H. Метод sequential_deepening

Последовательно проверяет текст с использованием различных методов. Если обнаружен подозрительный фрагмент, выполняет итеративное уточнение. Возвращает итоговый результат. Метод последовательного углубления проводит более глубокий анализ текста в случае обнаружения подозрительных фрагментов. Новизна метода состоит в его способности более детально исследовать подозрительные фрагменты текста для выявления скрытых или сложных угроз. Его достоинства включают обеспечение более глубокого и детального исследования текста для выявления угроз безопасности.

I. Метод detect_syntax_features

Метод анализа синтаксических признаков использует синтаксические шаблоны для обнаружения подозрительных фрагментов в тексте. Он анализирует зависимости между словами и их ролями в предложениях, ища определенные синтаксические шаблоны, такие как «subjpass», «aux», «neg», «acomp». Новизна метода заключается в его способности выявлять нестандартные структуры предложений, которые могут свидетельствовать о подозрительных фрагментах текста. Его преимущества включают способность выявления скрытых угроз, связанных с особенностями синтаксических конструкций.

J. Методы combination_1, combination_2, combination_3, combination_4, combination_5

Различные комбинации результатов методов для дополнительного контроля.

На рисунке 2 отобразим взаимосвязь между предлагаемыми методами обнаружения промпт-инъекции.

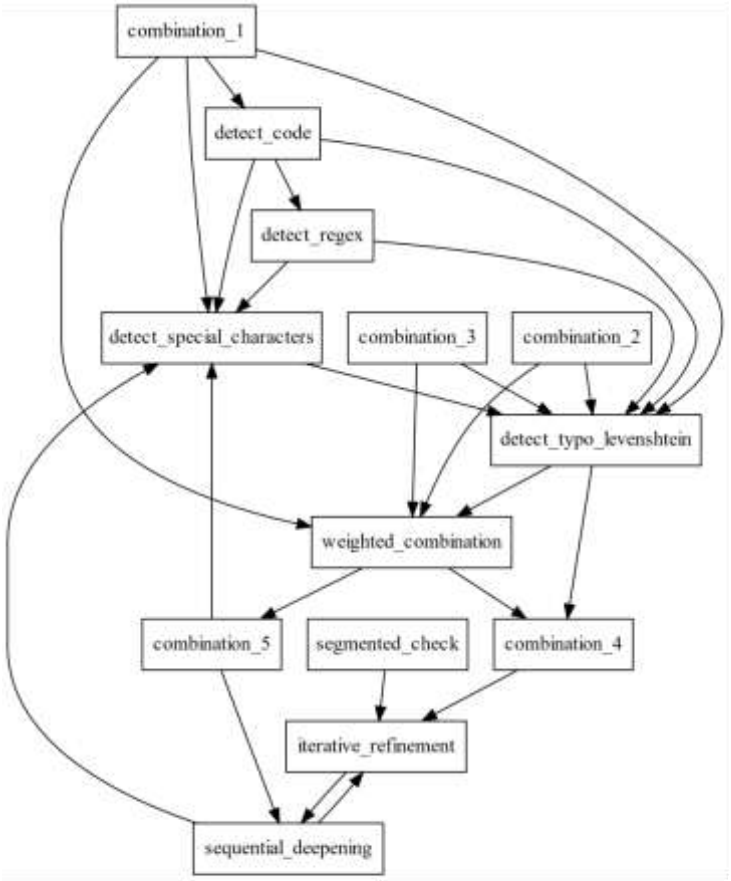

Рис. 2: Граф взаимосвязей применяемых методов обнаружения промпт-инъекций

Следует отметить, что после реализации данных методов защиты типа промпт-инъекции применялась классификация с помощью нейронной сети. Нейронная сеть, реализованная в данной задаче, представляет собой модель прямого распространения (feedforward neural network) [20]. Эта сеть предназначена для обучения на текстовых данных с целью детектирования атак типа «промпт-инъекции» на большие языковые модели. Рассмотрим основные аспекты использования, обучения и интерпретации данной нейронной сети [18].

На рисунке 3 представлен алгоритм работы разработанного ПО.

Алгоритм обнаружения промпт-инъекций начинает работу с анализа текста, который поступает на вход. В случае обнаружения признаков атаки алгоритм завершает свою работу без дальнейшей обработки текста. Если атака не обнаружена, алгоритм проводит дополнительный анализ для классификации текста как безопасного или подозрительного. После классификации безопасные тексты передаются для дальнейшей обработки моделью LLM, тогда как подозрительные тексты игнорируются.

Рис. 3: Алгоритм работы разработанного ПО

Модель состоит из трех полносвязных слоев (Dense layers), включая входной слой, один скрытый слой и выходной слой. Активационные функции ReLU применяются для внутренних слоев, а на выходном слое используется сигмоидальная активация, поскольку решается задача бинарной классификации (внедрен или нет вводимый текст) (рисунок 4).

Обучение нейронной сети включает в себя подготовку данных, построение модели, компиляцию модели с выбранным оптимизатором и функцией потерь, а затем обучение сети на обучающем наборе. В данной задаче производится обучение для каждого метода анализа, чтобы оценить их эффективность.

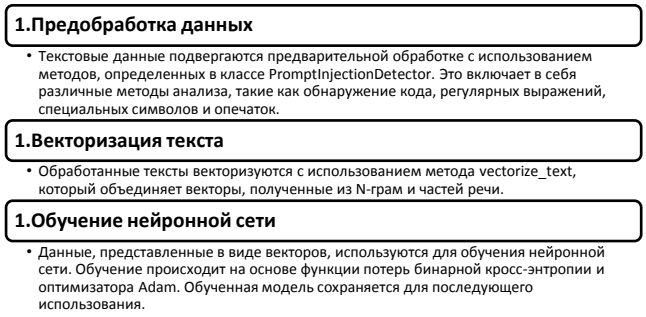

Рис. 4: Использование нейронной сети

Интерпретация результатов может осуществляться следующим образом:

1) Точность (Accuracy): Оценка процента правильных предсказаний модели.
2) Кривая ROC и AUC: Оценка качества классификации в зависимости от порога классификации.
3) Матрица ошибок (Confusion Matrix): Позволяет оценить количество верных и ложных положительных и отрицательных предсказаний.

В контексте атак типа «промпт-инъекции» также важно анализировать ложные срабатывания, чтобы минимизировать негативные воздействия на обычные тексты.

Этот подход к обучению и интерпретации нейронной сети позволяет создать модель, способную обнаруживать подозрительные фрагменты в текстах и принимать решение на основе различных методов анализа.

Также стоит отметить, что представленная методология реализуется на данных deepset/prompt-injections, представленном на Hugging Face [43].

V. Результаты исследования

По итогу тестирования разработанного алгоритма были получены следующие результаты (таблица 1):

Таблица 1: Сравнение качества моделей

Модель	Accuracy on test
1. Обнаружение кода	0,83
2. Регулярные выражения	0,86
3. Обнаружение специальных символов	0,83
4. Поиск опечаток с использованием расстояния Левенштейна	0,85
5. Взвешенная комбинация результатов	0,84
6. Сегментированная проверка	0,88
7. Итеративное уточнение	0,83
8. Последовательное углубление	0,86
9. Обнаружение синтаксических признаков	0,83
10. Комбинация 1 (обнаружение кода + обнаружение специальных символов)	0,87
11. Комбинация 2 (поиск опечаток с использованием расстояния Левенштейна + регулярные выражения)	0,89
12. Комбинация 3 (взвешенная комбинация результатов + анализ синтаксических признаков)	0,87
13. Комбинация 4 (сегментированная	0,86

проверка + итеративное уточнение)	
14. Комбинация 5 (последовательное углубление + обнаружение специальных символов)	0,85

Исследование эффективности методов обнаружения промпт-инъекций на базе больших языковых моделей представило важные результаты, подтверждающие разносторонний характер исследуемых методов. В соответствии с представленными в таблице 1 данными, можно сделать следующие ключевые выводы.

Высокая точность традиционных методов: Методы, базирующиеся на традиционных приемах, таких как обнаружение кода, использование регулярных выражений и анализ специальных символов, демонстрируют выдающуюся точность в районе 0,88. Это подтверждает их эффективность в обнаружении промпт-инъекций.

1) Лучшие модели: Модели, которые показали наивысшую точность на тестовой выборке, включают "Комбинацию 2" (0,89), "Сегментированную проверку" (0,88) и "Комбинацию 3" (0,87). Эти модели демонстрируют высокий уровень эффективности в обнаружении и предотвращении атак типа промпт-инъекции.
2) Эффективность методов: Методы, такие как использование регулярных выражений, поиск опечаток с использованием расстояния Левенштейна и комбинация различных подходов, показали себя эффективными с точки зрения обнаружения вредоносного кода или внедрения специальных символов.
3) Роль анализа синтаксических признаков: Взвешенная комбинация результатов с анализом синтаксических признаков, несмотря на ожидания, продемонстрировала немного более низкую точность (0,84). Это подчеркивает необходимость более глубокого исследования влияния синтаксических аспектов на обнаружение промпт-инъекций.
4) Необходимость комплексного подхода: Некоторые из лучших результатов были достигнуты благодаря комбинации нескольких методов. Это подтверждает важность комплексного подхода к противодействию атакам, который включает в себя различные техники и стратегии.

В целом, результаты исследования свидетельствуют о значительном прогрессе в разработке методов противодействия промпт-инъекциям на основе больших языковых моделей. Высокие показатели точности традиционных и итеративных методов подчеркивают их значимость в обеспечении безопасности при использовании языковых моделей, в то время как комбинированные подходы открывают перспективы для более глубокого понимания и эффективного противодействия подобным атакам.

С кодом разработанного ПО можно ознакомиться по ссылке:
https://github.com/RinaRam/PromptInjectionDefense.

VI. Заключение

Разработанная программа предназначена для обнаружения и защиты от атак типа промпт-инъекция на основе больших языковых моделей. Программа включает в себя методы обнаружения подозрительных вставок кода, регулярных выражений, специальных символов, а также опечаток. Кроме того, в программе используются комбинированные подходы, такие как взвешенная комбинация результатов различных методов, сегментированная проверка, итеративное уточнение и последовательное углубление анализа. Уникальность работы заключается в том, что используемые методы представляют комплексный подход к обнаружению и защите от таких атак, включая сочетание различных признаков и методов анализа, чего не наблюдается в большинстве подобных исследований. Именно за счет комбинированного метода удаётся достичь высокого уровня точности модели.

Основные шаги работы разработанной программы включают в себя:

1. Подготовка текста: текст предварительно обрабатывается с использованием токенизации.
2. Построение и обучение улучшенной нейронной сети: нейронная сеть, основанная на модели BERT, строится и обучается на данных, включающих как текстовые данные, так и различные признаки, выявленные методами анализа.
3. Обнаружение подозрительных вставок: используются различные методы анализа, такие как проверка наличия кода, регулярных выражений, специальных символов и опечаток, а также их комбинации.
4. Защитные меры: в зависимости от результатов анализа принимаются соответствующие защитные меры, включая отклонение подозрительных вставок или дополнительное углубление анализа.
5. Оценка и вывод результатов: точность каждого метода обнаружения оценивается, а также выводится общая точность комбинированных методов.

Этот подход представляет собой современную практику в области защиты от атак на базе машинного обучения и языковых моделей. Программа реализована на языке Python с использованием библиотек TensorFlow, Hugging Face Transformers и spaCy. Код системы доступен как для непосредственного использования, так и для дальнейшей разработки.

Полученные нами результаты свидетельствуют о значительном прогрессе в разработке методов противодействия промпт-инъекциям на основе больших языковых моделей. Высокие показатели точности традиционных и итеративных методов подчеркивают их значимость в обеспечении безопасности при использовании языковых моделей, в то время как комбинированные подходы предоставляют перспективы для более глубокого понимания и эффективного противодействия подобным атакам.

Разработанное в ходе исследований открытое программное обеспечение является общедоступным.

Получены конкурентоспособные метрики оценки качества для задачи обнаружения атак типа Prompt Injection на LLM. И, в дальнейшем, модель можно будет обучить на датасете на русском языке.

Благодарности

Авторы благодарны сотрудникам лаборатории Открытых информационных технологий кафедры Информационной безопасности факультета ВМК МГУ имени М.В. Ломоносова за обсуждения и ценные замечания.

Статья продолжает серию публикаций, начатых работой об обосновании исследований, посвященных устойчивым моделям машинного обучения [38]. Все публикации в журнале INJOIT, связанные с цифровой повесткой, начинались с работ В.П. Куприяновского и его соавторов [39-41]

Библиография

[1] Liu, Yupei, et al. "Prompt injection attacks and defenses in llm-integrated applications." arXiv preprint arXiv:2310.12815 (2023).

[2] Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran . 2023. A Survey of Adversarial Defences and Robustness in NLP. 1, 1 (April 2023), 43 pages. https://arxiv.org/pdf/2203.06414.pdf.

[3] Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. Безопасность систем машинного обучения. Large Language Models in Cybersecurity: State-of-the-Art //arXiv e-prints. – 2024. – С. arXiv:2402.00891.

[4] Reshabh K Sharma, Vinayak Gupta, and Dan Grossman. SPML: A DSL for Defending Language Models Against Prompt Attacks School of Computer Science & Engineering //arXiv e-prints. – 2024. – С. arXiv:2402.11755.

[5] Миттал А. Уязвимости и угрозы безопасности, с которыми сталкиваются большие языковые модели //Artificial Intelligence. unite.ai – 2024.

[6] Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kıcıman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models //arXiv e-prints. – 2024. – С. arXiv:2312.14197.

[7] Shayegani, Erfan, et al. "Survey of vulnerabilities in large language models revealed by adversarial attacks." arXiv preprint arXiv:2310.10844 (2023).

[8] Сухомлин В. А. и др. Модель навыков кибербезопасности 2020 //Современные информационные технологии и ИТ-образование. – 2020. – Т. 16. – №. 3. – С. 695-710.телекоммуникаций в науке и образовании. – 2021. – С. 452-456.

[9] Chen X. et al. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction //Proceedings of the ACM Web conference 2022. – 2022. – С. 2778-2788.

[10] Greshake K. et al. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models //arXiv e-prints. – 2023. – С. arXiv: 2302.12173.

[11] Hertz A. et al. Prompt-to-prompt image editing with cross attention control //arXiv preprint arXiv:2208.01626. – 2022.

[12] KV S., Manjunath T. C. Implementation of Authorization and Authentication techniques in IoT objects for Industrial Applications //International Neurourology Journal. – 2023. – Т. 27. – №. 4. – С. 500-509.

[13] Martin E. B., Ghosh S. GitHub Copilot: A Threat to High School Security? Exploring GitHub Copilot's Proficiency in Generating Malware from Simple User Prompts //2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC). – IEEE, 2023. – С. 1-6.

[14] Martínez Torres J., Iglesias Comesaña C., García-Nieto P. J. Machine learning techniques applied to cybersecurity //International Journal of Machine Learning and Cybernetics. – 2019. – Т. 10. – С. 2823-2836.

[15] Martino A., Iannelli M., Truong C. Knowledge injection to counter large language model (LLM) hallucination //European Semantic Web Conference. – Cham : Springer Nature Switzerland, 2023. – С. 182-185.

[16] Sarker I. H. Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective //SN Computer Science. – 2021. – Т. 2. – №. 3. – С. 154.

[17] Shulha O. et al. Banking information resource cybersecurity system modeling //Journal of Open Innovation: Technology, Market, and Complexity. – 2022. – Т. 8. – №. 2. – С. 80.

[18] Yan J. et al. Backdooring instruction-tuned large language models with virtual prompt injection //NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly. – 2023.

[19] Ye H. et al. Ontology-enhanced Prompt-tuning for Few-shot Learning //Proceedings of the ACM Web Conference 2022. – 2022. – С. 778-787.

[20] Zhuang L., Fei H., Hu P. Knowledge-enhanced event relation extraction via event ontology prompt //Information Fusion. – 2023. – Т. 100. – С. 101919.

[21] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In NeurIPS, 2020.

[22] Introducing ChatGPT. https://openai.com/blog/chatgpt, Retrieved: Mar, 2024.

[23] OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[24] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, et al. Palm 2 technical report. arXiv preprint arXiv:2305.10403, 2023.

[25] Bing Search. https://www.bing.com/, Retrieved: Mar, 2024.

[26] ChatGPT Plugins. https://openai.com/blog/chatgpt-plugins, 2023.

[27] ChatWithPDF. https://gptstore.ai/plugins/chatwithpdf-sdan-io, 2023.

[28] Sundar Pichai. An important next step on our AI journey. https://blog.google/technology/ai/bard-google-ai-search-updates/, Retrieved: Mar, 2024.

[29] Tiago A. Almeida, Jose Maria Gomez Hidalgo, and Akebo Yamakami. Contributions to the study of sms spam filtering: New collection and results. In Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11), 2011.

[30] Rich Harang. Securing LLM Systems Against Prompt Injection. https://developer.nvidia.com/blog/securing-llm-systemsagainst-prompt-injection, 2023.

[31] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. arXiv preprint arXiv:2306.05499, 2023.

[32] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. In NeurIPS ML Safety Workshop, 2022.

[33] Jose Selvi. Exploring Prompt Injection Attacks. https://research.nccgroup.com/2022/12/05/exploring-prompt-injection-attacks/, 2022.

[34] Simon Willison. Prompt injection attacks against GPT-3. https://simonwillison.net/2022/Sep/12/prompt-injection/, Retrieved: Mar, 2024.

[35] Simon Willison. Delimiters won't save you from prompt injection. https://simonwillison.net/2023/May/11/delimiters-wont-save-you, Retrieved: Mar, 2024.

[36] Hezekiah J. Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias,Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. arXiv preprint arXiv:2209.02128, 2022.

[37] Davey Winder. Hacker Reveals Microsoft's New AI-Powered Bing Chat Search Secrets. https://www.forbes.com/sites/daveywinder/2023/02/13/hacker-reveals-microsofts-new-ai-powered-bing-chat-search-secrets/?sh=356646821290, Retrieved: Mar, 2024.

[38] Намиот, Д. Е. Основания для работ по устойчивому машинному обучению / Д. Е. Намиот, Е. А. Ильюшин, И. В. Чижов // International Journal of Open Information Technologies. – 2021. – Т. 9, № 11. – С. 68-74. – EDN BAFFGK.

[39] Цифровая экономика и Интернет Вещей - преодоление силоса данных / В. П. Куприяновский, А. Р. Ишмуратов, Д. Е. Намиот [и др.] // International Journal of Open Information Technologies. – 2016. – Т. 4, № 8. – С. 36-42. – EDN WFVAPB.

[40] Искусственный интеллект как стратегический инструмент экономического развития страны и совершенствования ее государственного управления. Часть 2. Перспективы применения искусственного интеллекта в России для государственного управления / И. А. Соколов, В. И. Дрожжинов, А. Н. Райков [и др.]

// International Journal of Open Information Technologies. – 2017. – Т. 5, № 9. – С. 76-101. – EDN ZEQDMT.

[41] Цифровая экономика = модели данных + большие данные + архитектура + приложения? / В. П. Куприяновский, Н. А. Уткин, Д. Е. Намиот, П. В. Куприяновский // International Journal of Open Information Technologies. – 2016. – Т. 4, № 5. – С. 1-13. – EDN VWANDZ.

[42] Greshake, Kai, 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.

[43] Prompt injection dataset https://huggingface.co/datasets/deepset/prompt-injections Retrieved: Mar, 2024

Countering Prompt Injection attacks on large language models

Ramina Mudarova, Dmitry Namiot

Abstract - **Machine learning models have brought with them a new class of cyber attacks: adversarial attacks. Large language models are no exception and are also susceptible to attacks. Such attacks are becoming increasingly dangerous in the context of the use of deep learning and artificial intelligence in various fields. In the world of modern computing and artificial intelligence, security plays a key role and more and more attention is being paid to countering such attacks. Attacks on large language models include, but are not limited to, runtime attacks known as Prompt Injection. These attacks aim to disrupt large language models by injecting malicious instructions or prompts to corrupt the model's output, which can have serious consequences for the confidentiality and integrity of information. Technically, they turn out to be one of the easiest for attackers to execute. In this regard, there is a need to research and develop effective strategies to counter Prompt Injection. This article is devoted to the research and development of effective algorithms and methodologies capable of detecting and blocking Prompt Injection attacks in order to improve system security and protection from malicious influences. The key goal of the work is to implement these methods in the form of software solutions, as well as evaluate their effectiveness through experiments using various metrics on test data. The scientific novelty of this development lies in the creation of unique protection mechanisms that can ensure reliable security of language models from Prompt Injection attacks.**

Keywords - **Prompt Injection, large language models, machine learning methods, attacks on NLP models.**

REFERENCES

[1] Liu, Yupei, et al. "Prompt injection attacks and defenses in llm-integrated applications." arXiv preprint arXiv:2310.12815 (2023).
[2] Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran . 2023. A Survey of Adversarial Defences and Robustness in NLP. 1, 1 (April 2023), 43 pages. https://arxiv.org/pdf/2203.06414.pdf.
[3] [3]Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. Безопасность систем машинного обучения. Large Language Models in Cybersecurity: State-of-the-Art //arXiv e-prints. – 2024. – P. arXiv:2402.00891.
[4] Reshabh K Sharma, Vinayak Gupta, and Dan Grossman. SPML: A DSL for Defending Language Models Against Prompt Attacks School of Computer Science & Engineering //arXiv e-prints. – 2024. – P. arXiv:2402.11755.
[5] Mittal A. Vulnerabilities and security threats faced by large language models //Artificial Intelligence. unite.ai - 2024.
[6] Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kıcıman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models //arXiv e-prints. – 2024. – P. arXiv:2312.14197.
[7] Shayegani, Erfan, et al. "Survey of vulnerabilities in large language models revealed by adversarial attacks." arXiv preprint arXiv:2310.10844 (2023).
[8] Suhomlin V. A. i dr. Model' navykov kiberbezopasnosti 2020 //Sovremennye informacionnye tehnologii i IT-obrazovanie. – 2020. – T. 16. – #. 3. – S. 695-710.telekommunikacij v nauke i obrazovanii. – 2021. – S. 452-456.
[9] Chen X. et al. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction //Proceedings of the ACM Web conference 2022. – 2022. – S. 2778-2788.
[10] Greshake K. et al. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models //arXiv e-prints. – 2023. – S. arXiv: 2302.12173.
[11] Hertz A. et al. Prompt-to-prompt image editing with cross attention control //arXiv preprint arXiv:2208.01626. – 2022.
[12] KV S., Manjunath T. C. Implementation of Authorization and Authentication techniques in IoT objects for Industrial Applications //International Neurourology Journal. – 2023. – T. 27. – #. 4. – S. 500-509.
[13] Martin E. B., Ghosh S. GitHub Copilot: A Threat to High School Security? Exploring GitHub Copilot's Proficiency in Generating Malware from Simple User Prompts //2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC). – IEEE, 2023. – S. 1-6.
[14] Martínez Torres J., Iglesias Comesaña C., García-Nieto P. J. Machine learning techniques applied to cybersecurity //International Journal of Machine Learning and Cybernetics. – 2019. – T. 10. – S. 2823-2836.
[15] Martino A., Iannelli M., Truong C. Knowledge injection to counter large language model (LLM) hallucination //European Semantic Web Conference. – Cham : Springer Nature Switzerland, 2023. – S. 182-185.
[16] Sarker I. H. Deep cybersecurity: a comprehensive overview from neural network and deep learning perspective //SN Computer Science. – 2021. – T. 2. – #. 3. – S. 154.
[17] Shulha O. et al. Banking information resource cybersecurity system modeling //Journal of Open Innovation: Technology, Market, and Complexity. – 2022. – T. 8. – #. 2. – S. 80.
[18] Yan J. et al. Backdooring instruction-tuned large language models with virtual prompt injection //NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly. – 2023.
[19] Ye H. et al. Ontology-enhanced Prompt-tuning for Few-shot Learning //Proceedings of the ACM Web Conference 2022. – 2022. – S. 778-787.
[20] Zhuang L., Fei H., Hu P. Knowledge-enhanced event relation extraction via event ontology prompt //Information Fusion. – 2023. – T. 100. – S. 101919.
[21] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In NeurIPS, 2020.
[22] Introducing ChatGPT. https://openai.com/blog/chatgpt, Retrieved: Mar, 2024.
[23] OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
[24] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, et al. Palm 2 technical report. arXiv preprint arXiv:2305.10403, 2023.
[25] Bing Search. https://www.bing.com/, Retrieved: Mar, 2024.
[26] ChatGPT Plugins. https://openai.com/blog/chatgpt-plugins, 2023.
[27] ChatWithPDF. https://gptstore.ai/plugins/chatwithpdf-sdan-io, 2023.
[28] Sundar Pichai. An important next step on our AI journey. https://blog.google/technology/ai/bard-google-ai-search-updates/, Retrieved: Mar, 2024.
[29] Tiago A. Almeida, Jose Maria Gomez Hidalgo, and Akebo Yamakami. Contributions to the study of sms spam filtering: New collection and results. In Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11), 2011.
[30] Rich Harang. Securing LLM Systems Against Prompt Injection. https://developer.nvidia.com/blog/securing-llm-systemsagainst-prompt-injection, 2023.
[31] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications. arXiv preprint arXiv:2306.05499, 2023.
[32] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. In NeurIPS ML Safety Workshop, 2022.

[33] Jose Selvi. Exploring Prompt Injection Attacks. https://research.nccgroup.com/2022/12/05/exploring-prompt-injection-attacks/, 2022.
[34] Simon Willison. Prompt injection attacks against GPT-3. https://simonwillison.net/2022/Sep/12/prompt-injection/, Retrieved: Mar, 2024.
[35] Simon Willison. Delimiters won't save you from prompt injection. https://simonwillison.net/2023/May/11/delimiters-wont-save-you, Retrieved: Mar, 2024.
[36] Hezekiah J. Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias,Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. arXiv preprint arXiv:2209.02128, 2022.
[37] Davey Winder. Hacker Reveals Microsoft's New AI-Powered Bing Chat Search Secrets. https://www.forbes.com/sites/daveywinder/2023/02/13/hacker-reveals-microsofts-new-ai-powered-bing-chat-search-secrets/?sh=356646821290, Retrieved: Mar, 2024.
[38] Namiot, D. E. Osnovanija dlja rabot po ustojchivomu mashinnomu obucheniju / D. E. Namiot, E. A. Il'jushin, I. V. Chizhov // International Journal of Open Information Technologies. – 2021. – T. 9, # 11. – S. 68-74. – EDN BAFFGK.
[39] Cifrovaja jekonomika i Internet Veshhej - preodolenie silosa dannyh / V. P. Kuprijanovskij, A. R. Ishmuratov, D. E. Namiot [i dr.] // International Journal of Open Information Technologies. – 2016. – T. 4, # 8. – S. 36-42. – EDN WFVAPB.
[40] Iskusstvennyj intellekt kak strategicheskij instrument jekonomicheskogo razvitija strany i sovershenstvovanija ee gosudarstvennogo upravlenija. Chast' 2. Perspektivy primenenija iskusstvennogo intellekta v Rossii dlja gosudarstvennogo upravlenija / I. A. Sokolov, V. I. Drozhzhinov, A. N. Rajkov [i dr.] // International Journal of Open Information Technologies. – 2017. – T. 5, # 9. – S. 76-101. – EDN ZEQDMT.
[41] Cifrovaja jekonomika = modeli dannyh + bol'shie dannye + arhitektura + prilozhenija? / V. P. Kuprijanovskij, N. A. Utkin, D. E. Namiot, P. V. Kuprijanovskij // International Journal of Open Information Technologies. – 2016. – T. 4, # 5. – S. 1-13. – EDN VWANDZ
[42] Greshake, Kai, 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.
[43] Prompt injection dataset https://huggingface.co/datasets/deepset/prompt-injections Retrieved: Mar, 2024.