# Извлечение тренировочных данных: Риски и решения в контексте безопасности LLM

Д.В. Герасименко, Д.Е. Намиот

*Аннотация*—**Качество результатов современных языковых моделей неотъемлемо связано с объемом данных, на которых эта модель будет обучена. Последние громкие расследования вокруг компаний в области искусственного интеллекта были как раз связаны с неправомерным использованием информации, которая была получена в сети интернет. Другой стороной борьбы за использование данных о пользователях является негласное расширение пользовательских соглашений, где компании разрешается использовать полученую информацию для обучения своих моделей. Данная работа посвящена анализу современных проблем, связанных с извлечением тренировочных данных из больших языковых моделей (LLM), таких как семейства GPT и Llama. Использование большого количества неструктурированных данных для обучения современных моделей делает эти модели привлекательными объектами для атак, направленных на получение доступа к этим данным или их характеристикам.**

**В статье освещается таксономия атак, направленных на извлечение тренировочных данных и описываются последствия, которые могут возникнуть от неправомерного использования языковых моделей. В результате исследования было показано, что без должной защиты тренировочные данные могут быть использованы злоумышленниками для восстановления конфиденциальной информации, которая в свою очередь ставит под угрозу не только пользователей, но и репутацию организаций.**

*Ключевые слова*—**большие языковые модели, извлечение данных, безопасность, конфиденциальность.**

## I. Введение

Современные большие языковые модели (large language model, LLM) представляют собой эволюцию и усовершенствование предыдущих моделей, таких как семейства GPT или Llama, для обработки естественного языка. Они обладают огромным количеством параметров и способны генерировать текст, отвечать на вопросы, анализировать тональность текста, проводить машинный перевод и многое другое с высокой точностью и натуральностью. Однако стоит учитывать, что ключевой проблемой при обучении LLM остается выбор набора данных, на которых модель будет обучаться. В частности, мы будем обращать внимание на качество предоставляемых данных и потенциальные риски от их неправомерного использования.

Прежде всего стоит рассмотреть проблему качества данных. Обученные большие языковые модели могут содержать ошибки, несоответствия, предвзятость и другие недочеты, которые могут повлиять на эффективность модели и саму возможность ее применения. Можно выделить несколько ключевых аспектов качества данных в LLM:

- **Количественное качество данных**: объем данных, на которых обучается модель, играет важную роль в её качестве. Чем больше разнообразных данных используется для обучения, тем лучше модель способна обобщать и выполнять задачи.
- **Репрезентативность**: важно чтобы данные, использованные для обучения модели, были репрезентативными для задач, которые модель должна решать. Недостаточно разнообразные и несбалансированные данные могут привести к искажениям и ошибкам в выводах модели.
- **Качество разметки данных**: данные должны быть хорошо размечены и качественно подготовлены для обучения модели. Некорректная разметка может привести к неверным выводам и ошибкам, поэтому важно уделить внимание этому аспекту.
- **Чистота данных**: Наличие шума или искажений в данных может негативно сказаться на качестве модели. Поэтому необходимо проводить предварительную обработку данных, удаление дубликатов, коррекцию ошибок и другие меры для повышения чистоты данных.

Кроме того, важно учитывать этические угрозы и риски в случае реализации атаки на извлечение данных из больших языковых моделей. Это означает, что необходимо обеспечивать соблюдение конфиденциальности и защиты личных данных, избегать использования информации в недобросовестных целях и учитывать потенциальные негативные последствия использования таких данных, например, нарушение авторских прав или искажение существующих фактов и стереотипов. Исходя из вышесказанного, разработчики и исследователи должны принимать все необходимые меры для обеспечения качества, надежности и безопасности используемых данных, строго соблюдать нормы этики и правовые требования. Примером подобных требований может являться Регламент об искусственном интеллекте, который классифицирует продукты и приложения с использованием ИИ на три категории: запрещенные системы (с недопустимым уровнем риска), системы с высокой степенью риска и остальные системы искусственного интеллекта. Принимая во внимание требования регуляторных организаций, проблема извлечения обучающих данных из LLM становится все более актуальной в связи с растущим интересом к исследованиям в области искусственного интеллекта и машинного обучения. Большие языковые модели, такие как GPT, BERT или Llama требуют не

Статья получена 30 сентября 2024.

Денис Валерьевич Герасименко, МГУ им. М.В. Ломоносова, (email: rsu.deger@gmail.com).

Дмитрий Евгеньевич Намиот, МГУ им. М.В. Ломоносова, (email: dnamiot@gmail.com

только огромного объема данных для обучения, но и сталкиваются с рядом проблем, когда речь идет о сборе данных, включая следующие:

1. **Конфиденциальность и защита данных**: большие языковые модели могут содержать конфиденциальные и личные данные, которые могут быть неправомерно использованы, если попадут не в те руки. Результатом подобных действий может стать, например, учетка персональных данных.
2. **Сохранение контекста**: извлечение обучающих данных из языковых моделей может привести к раскрытию информации о контексте, в рамках которого происходило обучение. Это может привести к распространению конфиденциальной информации.
3. **Подмена контента**: в некоторых случаях сбор данных из больших языковых моделей может привести к созданию контента или информации, которые могут быть введены в оборот как аутентичные, хотя на самом деле они сгенерированы искусственным путем.
4. **Этические аспекты**: важно также учитывать этические аспекты при извлечении данных из больших языковых моделей. Это включает в себя вопросы объективности, предвзятости и возможного использования моделей для манипуляции информацией.

В данной статье предлагается рассмотреть существующие векторы атак на извлечение тренировочных данных из LLM, как существующие современные средства защиты влияют на безопасность тренировочных данных и какие инструменты с открытым исходным кодом могут быть использованы для тестирования языковых моделей. Статья является частью серии публикаций, написанных для поддержки магистерской программы факультета ВМК МГУ имени М.В. Ломоносова. Представленная статья имеет следующую структуру. В разделе II будет рассмотрена таксономия атак. Раздел III посвящен атакам на извлечение тренировочных данных по побочным каналам. В разделе IV представлены практические результаты атак на семейство моделей GPT. И раздел V содержит заключение.

## II. Таксономия атак на извлечение тренировочных данных

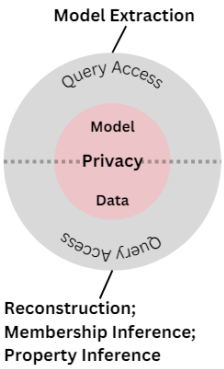


Рис. 1. Таксономия атак извлечения тренировочных данных.

В качестве источника таксономии был выбран документ Национального института стандартов и технологий (NIST): Adversarial Machine Learning. A Taxonomy and Terminology of Attacks and Mitigations [1]. Он позволяет взглянуть на классификацию атак на извлечение тренировочных данных в разрезе большого количества систематизированных научных статей и работ видных исследователей в области машинного обучения. В частности, проблемы конфиденциальности используемых данных давно являются предметом обеспокоенности, а описание самих атак на конфиденциальность в отношении агрегированной информации, собранной из пользовательских записей, началось с фундаментальной работы Динура и Ниссима об атаках, которые были нацелены на восстановление данных [2]. Их целью является восстановление конфиденциальной информации о пользователе или получение чувствительных данных критической инфраструктуры (например, приватные ключи, токены или документы) из общего доступа к агрегированной информации. Другая атака на конфиденциальность – это атака на информацию о принадлежности, в которой злоумышленник может определить, содержится ли определенная запись в наборе данных, используемом для вычисления статистической информации или обучения выбранной модели. В современной литературе основное внимание уделяется атакам на принадлежность в основном в условиях «черного ящика», в которых злоумышленники имеют возможность отправлять большое количество запросов к обученной модели машинного обучения. Более того, современная классификация позволяет рассматривать атаки на извлечение тренировочных данных независимо от используемой техники атаки и ее конечной цели – это может быть восстановление данных, запоминание тренировочных данных, атаки на принадлежность, извлечение модели и вывод ее свойств или возникающие побочные каналы, которые используют при попытках усилить безопасность используемой модели. Далее я хочу остановиться на некоторых классах атак подробнее.

### *A. Восстановление (реконструкция) данных*

Атаки на восстановление данных вызывают наибольшее беспокойство с точки зрения конфиденциальности, поскольку они способны восстановить данные отдельного лица из опубликованной агрегированной информации. Опубликованные работы показывают, что исходная атака требует экспоненциального количества запросов для восстановления, но последующие работы показали, как можно выполнить восстановление с полиномиальным количеством запросов, например в работе “Exposed! A Survey of Attacks on Private Data” [3] показано, как подобная атака берет, казалось бы, безобидную опубликованную информацию и использует ее для выявления личных данных отдельных лиц, тем самым демонстрируя, что такая информация ставит под угрозу конфиденциальность данных. Например, атаки с повторной идентификацией показали, что легко связать якобы обезличенные записи с личностью заинтересованного лица.

В контексте классификаторов ML были также представлены успешные атаки, которые позволяют определять участников класса из обучающих данных модели. Недавние исследования в работе “Reconstructing training data with informed adversaries” [4] показывают, что возможно обучить сеть-реконструктор, которая сможет вос-

станавливать часть выборки данных из модели нейронной сети, предполагая, что сеть-реконструктор получает на входе веса атакуемой модели.

Резюмируя, способность восстанавливать обучающие образцы частично объясняется тенденцией нейронных сетей запоминать свои тренировочные данные. В исследуемых работах было показано, что запоминание меток обучения необходимо для достижения почти оптимальной обобщающей ошибки в машинном обучении.

### *B. Выводы о принадлежности*

Атаки на определение принадлежности также раскрывают личную информацию о человеке, как и атаки на восстановление или запоминание, и все еще вызывают большую озабоченность в случае публикации агрегированной информации или использовании моделей машинного обучения, обученных на пользовательских данных. В определенных ситуациях знание того, что человек является частью обучающего набора, уже имеет проблемы конфиденциальности, например, в медицинском исследовании пациентов с редким заболеванием. Более того, определение принадлежности может быть использовано в качестве предварительного этапа для проведения атак на извлечение тренировочных данных. При определении принадлежности целью атакующего является определить, была ли определенная запись или выборка данных частью обучающего набора, использованного для статистического или обучающего алгоритма машинного обучения. В течение последних пяти лет в литературе активно используется понятие ”атаки на принадлежность”при обсуждении угроз моделей машинного обучения. Большинство из таких атак описываются в контексте глубоких нейронных сетей, применяемых для задач классификации .Подобно другим атакам в области состязательного машинного обучения, определение принадлежности может быть реализовано в условиях белого ящика, где атакующие имеют знание об архитектуре и параметрах модели, но большинство атак разработаны для условий черного ящика, где атакующий генерирует запросы к обученной модели машинного обучения. [5]

Что касается методов проведения атак на определение принадлежности, атака на основе потерь, предложенная в работе “ Privacy risk in machine learning: Analyzing the connection to overfitting” [6] является одним из наиболее эффективных и широко используемых методов. Используя знание о том, что модель машинного обучения минимизирует потери на обучающих образцах, атака определяет, что целевой образец является частью обучения, если его потери ниже фиксированного порога (выбираемого как среднее потерь обучающих примеров). Другим популярным методом, предложенным в работе “ Membership inference attacks against machine learning models” [7] являются теневые модели, которые обучают мета-классификатор на примерах внутри и вне обучающего набора, полученных из тренировки тысяч теневых моделей машинного обучения для той же задачи, что и оригинальная модель. Этот метод обычно дорогостоящий и хотя он может показывать высокие результаты, его вычислительная стоимость высока и требует доступа к множеству образцов из распределения для обучения теневых моделей. Эти два метода находятся на противоположных полюсах спектра по сложности, но они проявляют сходную производительность в терминах точности при низких уровнях ложноположительных значений.

Стоит отметить, что уже существует несколько общедоступных библиотек, которые предлагают реализацию атак на основе вывода о принадлежности: TensorFlow Privacy library [8] и ML Privacy Meter [9].

### *C. Извлечение модели*

В современном мире все больше компаний приходят к тому, что выгоднее, безопаснее и быстрее работать в облаке. Это приводит к тому, что помимо стандартных услуг облачных провайдеров, таких как: **IaaS** (Infrastructure as a Service), **PaaS** (Platform as a Service) и **SaaS** (Software as a Service) появляется новое направление услуг – **MLaaS** (Machine Learning as a Service). Это концепция, при которой компании или разработчики могут использовать облачные платформы или сервисы сторонних провайдеров для доступа к мощности машинного обучения без необходимости владеть собственной инфраструктурой или глубокими знаниями в области машинного обучения.

В сценариях использования MLaaS облачные поставщики обычно обучают LLM, используя собственные данные, стараясь сохранить конфиденциальность архитектуры и параметров модели. Соответственно, цель злоумышленника при проведении атаки на извлечение модели заключается в извлечении информации об архитектуре модели и параметрах путем отправки запросов к модели машинного обучения, обученной поставщиком MLaaS. Первые атаки на кражу моделей были продемонстрированы в работе “ Stealing Machine Learning Models via Prediction APIs” [10] на нескольких онлайн-сервисах машинного обучения для различных моделей. Однако было показано, что точное извлечение моделей машинного обучения невозможно. Вместо этого может быть восстановлена функционально эквивалентная модель, отличная от оригинальной модели, но достигающая сходного качества результатов для выполнения задач предсказания.

В опубликованной литературе и статьях было представлено несколько методик для проведения атак на извлечение моделей. Первый метод - это прямое извлечение на основе математической формулировки операций, выполняемых в глубоких нейронных сетях, что позволяет атакующему алгебраически вычислить веса модели. Вторая техника, исследованная в ряде статей, заключается в использовании методов обучения для извлечения. Например, активное обучение может направлять запросы к атакуемой модели для более эффективного извлечения весов модели, а обучение с подкреплением будет стараться адаптировать стратегию, которая сокращает количество запросов. Третья техника - использование информации о ”побочных каналах”для извлечения модели, например атака rowhammer на динамическую оперативную память (DRAM).

Следует отметить, что извлечение моделей часто не является конечной целью, а лишь шагом к другим атакам. Поскольку веса и архитектура модели становятся известными, злоумышленники могут запустить более мощные атаки, характерные для ситуаций ”белого ящика”или

”серого ящика”. Поэтому предотвращение извлечения моделей может смягчить последующие атаки, зависящие от того, что у атакующего есть знание об архитектуре модели и ее весах.

### *D. Вывод свойств*

В атаках на вывод свойств злоумышленник пытается узнать глобальную информацию о распределении обучающих данных, взаимодействуя с моделью машинного обучения. Например, злоумышленник может определить долю обучающего набора данных с определенным чувствительным атрибутом, таким как демографическая информация, что может раскрыть потенциально конфиденциальную информацию о наборе данных, которая не предназначалась для раскрытия. На данный момент атаки на вывод свойств были разработаны как для концепции белого ящика, где злоумышленник имеет доступ ко всей ML модели, так и для чёрного ящика, где злоумышленник выставляет запросы к модели и узнаёт либо предсказанные метки, либо вероятности классов. [11], [12], [13] Эти атаки были описаны и продемонстрированы для скрытых марковских моделей, метода опорных векторов, нейронных сетей с прямой связью, сверточных нейронных сетей, моделей распределенного обучения, генеративно-состязательных сетей и графовых нейронных сетей.

### *E. Митигация*

Изучение атак на восстановление данных сподвигло к строгому определению дифференциальной приватности (ДП). Дифференциальная приватность — это чрезвычайно сильный набор методов усиления конфиденциальности, гарантирующие ограничение того, насколько злоумышленник, имеющий доступ к результатам алгоритма, может узнать о каждой отдельной записи в наборе данных. Исходное чистое определение ДП имеет параметр конфиденциальности $\epsilon$ (т.е. коэффициент приватности), который ограничивает вероятность того, что злоумышленник, имеющий доступ к выходу алгоритма, может определить, включена ли определенная запись в набор данных. ДП была расширена до понятий приближенной ДП, которая включает второй параметр $\delta$, который интерпретируется как вероятность случайной утечки информации в дополнение к $\epsilon$.

ДП стала широко использоваться из-за нескольких полезных свойств: конфиденциальности группы (т.е. расширение определения до двух наборов данных, отличающихся в k записях), постобработки (т.е. конфиденциальность сохраняется даже после обработки вывода) и композиции (т.е. конфиденциальность составляется, если выполняются несколько вычислений на наборе данных). Самым широко используемым алгоритмом ДП для обучения моделей машинного обучения является DP-SGD, с недавними улучшениями, такими как DP-FTRL и ДП с матричным разложением.

По определению ДП обеспечивает защиту от атак на восстановление данных и вывод участия. Фактически, определение ДП предполагает наличие верхней границы успеха для злоумышленника в проведении атак вывода принадлежности. Однако ДП не предоставляет гарантий против атак на извлечение модели, поскольку этот метод предназначен для защиты тренировочных данных, а не модели. В нескольких работах описываются негативные результаты использования дифференциальной приватности для защиты от атак на вывод свойств, которые были направлены на извлечение свойств подгрупп в обучающем наборе данных. При этом одной из основных проблем использования ДП на практике является настройка параметров приватности для достижения компромисса между уровнем конфиденциальности информации и достигнутой полезности модели. Анализ алгоритмов, обеспечивающих конфиденциальность, таких как DP-SGD, часто показывает посредственные результаты и не является обязательным при настройке параметров конфиденциальности в используемой модели, так как часто приводит к ухудшению качества работы.

## III. Атаки на извлечение тренировочных данных по побочным каналам

Рассмотренные выше классы атак на извлечение тренировочных данных предполагают, что модели существуют в вакууме, тогда как на самом деле модели ML являются частью более крупных систем, которые включают компоненты для фильтрации обучающих данных, мониторинга вывода и других элементов. Эти компоненты широко внедрены в реальные системы ML для повышения точности, безопасности и робастности. В этом разделе рассмотрены побочные каналы, которые возникают при использовании подобных методов защиты и задействуют компоненты системы для извлечения конфиденциальной информации с гораздо более высокой скоростью, чем это возможно для автономных моделей.

Опираясь на работу “Privacy Side Channels in Machine Learning Systems” [14] можно сделать вывод, что следующие четыре категории атак, охватывают весь жизненный цикл ML:

- **Фильтрация обучающих данных**. Большинство крупных датасетов фильтруются для удаления дубликатов и аномальных примеров. В исследованиях было продемонстрировано, что фильтры данных создают побочные каналы, поскольку они создают зависимости между данными разных пользователей. В свою очередь, злоумышленники могут усилить атаки на конфиденциальность, вставляя “отравленные” примеры, которые максимизируют эти зависимости. Например, дедупликация данных - техника, разработанная для улучшения конфиденциальности, может ухудшить конфиденциальность, даже приводя к нарушениям гарантий дифференциальной приватности (ДП).
- **Предварительная обработка ввода**. Многие модели требуют предварительной обработки своих входных данных, например, языковые модели требуют токенизации текста. Это может создавать побочные каналы, которые позволяют злоумышленникам извлекать конфиденциальную информацию, такую как редкие слова в обучении.
- **Фильтрация вывода модели**. Для улучшения конфиденциальности многие системы машинного обучения используют фильтры, которые предотвращают вывод моделями реальных данных обучения. Это может создать побочный канал, который фактически уменьшает конфиденциальность настолько, что

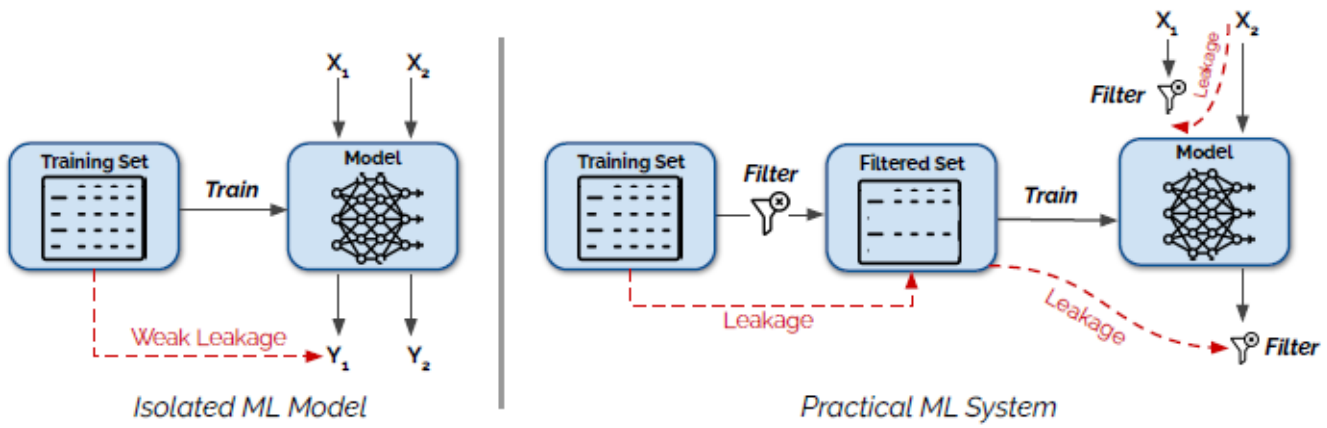


Рис. 2. Ориентировочный вид пайплайна ML для изолированной и вероятной модели в производственной среде.

становится возможной практически идеальная атака вывода о принадлежности.

- **Фильтрация запросов**. Многие системы машинного обучения используют фильтры для запросов во время тестирования, которые отклоняют определенные входные данные, например, детекторы атакующих примеров и атаки на извлечение модели. Поскольку многие из этих фильтров агрегируют информацию по различным пользователям (для защиты от атаки Сивиллы), злоумышленники могут раскрывать информацию о запросах других пользователей, отправляя таргетированные входные данные.

В совокупности, приведенные побочные каналы подсвечивают проблему тестирования безопасности ML в изолированной среде и необходимость системного подхода к этому вопросу. Стоит учитывать, что описанные выше побочные каналы не используют такие физические явления как изменение потребления энергии, побочное электромагнитное излучение (ПЭМИН) и переключение контекста GPU, через которые тоже возможно восстановить веса или архитектуру отдельных моделей машинного обучения. Описанные далее атаки не потребуют никакого физического доступа или тонкой детализированной измерительной информации о модели машинного обучения. Будет достаточно только доступа в виде черного ящика к функциям предсказания системы.

### *A. Атака на фильтры тренировочных данных*

Большинство LLM тренируются на данных, извлеченных из интернета. Из-за характера контента в интернете часто бывает, что некоторые тренировочные данные повторяются множество раз и это проблематично по нескольким причинам, включая то, что повторяющиеся тренировочные данные намного вероятнее запоминаются. Процесс дедупликации решает эту проблему путем удаления точных или близко дублирующихся образцов.

**Побочный канал**. Дедупликация данных вводит взаимозависимости между точками данных: образец x удаляется только в случае, если некий похожий образец x′ присутствует в обучающем наборе. Атакующий может использовать этот побочный канал для осуществления целенаправленной атаки на вывод о принадлежности. Предположим, что атакующий хочет вывести членство некоторого образца x и знает, что некоторые близкие дубликаты x′ находятся в Dtrain. Основываясь на этом, атакующий может сделать вывод, находится ли x в Dtrain, определив, были ли удалены дубликаты x′ или нет. По сравнению со стандартной атакой вывода о принадлежности на целевую точку x, этот побочный канал усиливает утечку конфиденциальности, если дубликаты x′, известные атакующему, будут запомнены моделью с большей вероятностью, чем оригинальный образец.

**Атака**. Исходим из того, что у атакующего есть возможность отравить небольшую часть датасета, чтобы подсветить информацию о других данных. В приведенном приме атакующий отравляет тренировочный датасет, вставляя дубликаты x′ целевой точки данных x. Предположим, что атакующий имеет возможность обращаться к обученной модели в парадигме черного ящика, знаком с используемой процедурой дедупликации данных и сама дедупликация не зависит от метки образца, т.е. две точки (x, y) и (x′, y′) являются дубликатами, если x близок к x′, даже если y $\neq$ y′. Данные условия приводят к тому, что появляется 4 пути для атаки на вывод о принадлежности при использовании дедупликации:

*Точная дедупликация и полное удаление*: при заданной цели (x, y) добавляется дубликат с неверным меткой (x, y′) в набор данных. Затем делается вывод о непринадлежности дублированной точки: если эта точка отсутствует, мы знаем, что цель присутствовала до дедубликации. Для этой атаки достаточно использовать всего один пример отравления.

*Точная дедупликация и удаление всех, кроме одного*: в этом случае мы по-прежнему используем ту же атаку, что и выше, но она менее сильная: если цель явно принадлежит классу, то отравленный дубликат удаляется только с вероятностью 50%. Для этой атаки также достаточно использовать один пример отравления.

*Неточная дедупликация и удаление всех, кроме одного*: это наиболее сложный и интересный сценарий атаки. Мы создаем N приблизительных дубликатов ($x'_1$, y′), . . . , ($x'_n$, y′) в виде звездообразной топологии (Hub-and-Spoke, Рис. 3). Это делает дубликаты атакующего близкими к цели, т.е. sim(x, $x'_i$) $\geq$ α для всех i, но не близкими друг к другу, т.е. sim($x'_i$, $x'_j$) < $\alpha$ для всех i $\neq$ j. Затем злоумышленник запускает атаку на вывод о не принадлежности по всем N отравленным образцам $x'_i$. Таким образом, атака становится сильнее с введением

большего числа примеров отравления.

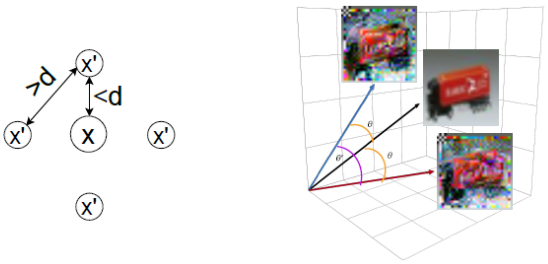


Рис. 3. Слева: отравленные примеры близки к центру (X), но далеки друг от друга. Справа: вывод изображений при атаке.

*Неточная дедупликация и полное удаление*: Мы повторяем ту же атаку неточной дедубликации, но атака становится сильнее, по мере удаления всех примеров.

**Результат**. Дедупликация создает сильный побочный канал, который обеспечивает практически идеальное заключение о принадлежности данных.

На представленном графике (Рис. 4) отражена сила атак на определение принадлежности для моделей обученных как с полным удалением данных при точной дедупликации, так и без. На дедуплицированных данных атака достигает практически идеального определения членства, а когда дедупликации данных не применяется, атака все равно превосходит LiRA из-за усиления, которое происходит при отравлении данных.

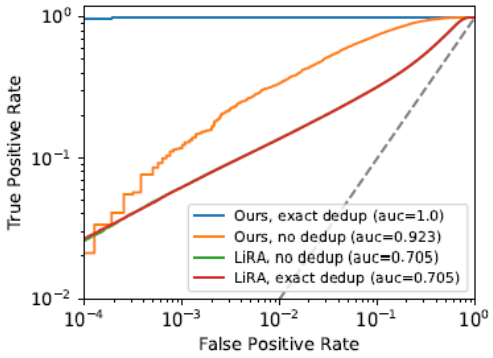


Рис. 4. Дедупликация может значительно ухудшить конфиденциальность. Представлена эффективность сужения членства как при точной дедупликация (удаление всех), так и без дедупликации.

### *B. Атака на входные и выходные фильтры*

В данном разделе будет рассмотрено, как фильтры, применяемые ко входам или выходам модели, могут обеспечить сильные атаки на определение принадлежности. Фактически, эти атаки на определение принадлежности могут быть настолько сильными, что их можно превратить в атаки на извлечение данных, которые позволяют даже извлекать закрытые ключи OpenSSH, содержащиеся в наборе тренировочных данных для обучения языковой модели. Основной предпосылкой атаки является то, что фильтры систем машинного обучения делают невозможным для системы генерацию определенных выходов в зависимости от используемых тренировочных данных. Атакующий может догадаться о входных данных в обучающем наборе и создать запрос, который приведет к невозможному выводу, если предположение окажется верным. Это дает идеальную атаку вывода без принадлежности: если модель выдает невозможный результат, атакующий знает, что его предположение неверно.

**Побочный канал**. Ограничивание входов до фиксированного размера окна позволяет атакующему извлекать весь словарь любой модели. В частности, рассмотрим последовательность вида «Мой любимый цвет - красный. Какой мой любимый цвет?». Здесь большинство языковых моделей предскажут слово ”красный”в качестве продолжения. Однако, если добавить какой-то текст-шаблон длиной $\geq$ N токенов, например, «Мой любимый цвет - красный. ТЕКСТ-ШАБЛОН. Какой мой любимый цвет?», то модель не сможет увидеть слово «красный» для длинных последовательностей подобных шаблонов. Это позволит определить, сколько токенов занимает последовательность входа модели.

**Атака 1**. В общем случае, использование алгоритма кодирования байт-пар, гарантирует, что любой токен может быть рекурсивно разделен на два подтокена до тех пор, пока конечные подтокены не станут состоять из одного байта. Описанная атака будет использовать этот алгоритм наоборот, инициализируя извлеченный словарь V всеми отдельными байтами и рекурсивно расширяя его. Тем самым, для всех пар токенов (u, v) $\in$ V $\times$ V, запрашивается модель-предложение:

$$\text{“My favorite color is red. } \underbrace{u||v\ u||v\ \ldots\ u||v}_{3N/4 \text{ times}} \text{ My favorite color is”}$$

где || обозначает конкатенацию. Предположим, что u||v действительно является одним токеном в словаре. Тогда количество вставленных шаблонных токенов между вопросом и ответом будет всего 3N/4, и модель ответит ”красный”. В этом случае мы добавляем новый токен V $\leftarrow$ V $\cup$ u||v. С другой стороны, если u||v был представлен двумя токенами, то количество шаблонных токенов будет 3N/2 > N, и модель не ответит ”красный”.

**Результат**. Описанная атака оказалась эмпирически эффективна при извлечении всего словаря модели GPT-2. Были извлечены редкие подстроки, такие как «RandomRedditorWithNo», «TheNitromeFan» и «SolidGoldMagikarp». Эти строки являются именами пользователей Reddit, которые, вероятно, часто повторялись в обучающем наборе данных для GPT-2. Атака потребовала 819 869 857 запросов.

**Атака 2**. Второй вариант атаки фокусируется уже на извлечении целевого словаря. Такая атака может быть полезна для генерации состязательных примеров, где может быть полезно знать токенизацию определенной строки. Чтобы использовать вышеуказанный алгоритм для конкретной строки, такой как «hello» и «world», мы можем просто проверить, являются ли «he», «el», «ll» и т. д. токенами и при этом не нужно проверять, являются ли «ho», «wl», «ol» и т. д. тоже токенами.

**Результат**. В приведенной таблице измеряется эффективность этой ограниченной атаки, сообщая количество запросов, необходимых для определения того, как GPT-2 токенизирует первые N байт набора данных enwiki8 (подмножество Википедии). В среднем требуется примерно 30 000 запросов на мегабайт, снижаясь по мере увеличения числа байт для токенизации.

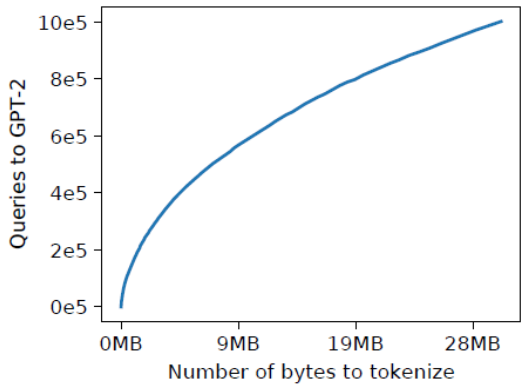


Рис. 5. Извлечение токенизатора для GPT-2 на байтовых строках из Википедии

### *C. Атака на фильтры запоминания*

LLM-модели известны своей способностью запоминать и выводить последовательности из своего обучающего набора данных. Поэтому популярные коммерческие системы с LLM, такие как Copilot, используют фильтры, которые блокируют выводы, совпадающие с обучающими данными. Таким образом, гарантируется, что модели никогда не выдают дословную последовательность из своего тренировочного набора данных. Подобное декодирование без запоминания добавляет фильтр, который работает онлайн с языковой моделью и, перед тем как выдать следующий токен, проверяет, вызовет ли это совпадение k-граммы с последовательностью из тренировочного набора данных. Если да, этот токен заменяется на наиболее вероятный пропущенный токен. Хотя такие фильтры недостаточно идеальны (например, они не предотвращают модель от вывода похожих, но не дословных копий обучающих данных), они представляют собой эффективную и практичную защиту, которая снижает вероятность копирования данных.

**Побочный канал**. Применение любой формы декодирования без запоминания приводит к значительной уязвимости конфиденциальности: если языковая модель когда-либо генерирует какую-либо особую k-грамму, мы гарантированно знаем, что она не была частью тренировочных данных. Это дает нам идеальную атаку на непринадлежность (т.е. стопроцентную истинную отрицательную оценку при нулевой ложной отрицательной оценке).

**Атака**. Некоторые модели содержат фильтры запоминания, которые применяются постоянно. Проблема с такими фильтрами заключается в том, что если модель не выводит некоторый суффикс, это может быть связано с тем, что фильтр был активирован (т. е. последовательность находится в обучающем наборе данных) или потому, что модель просто низко оценивает вероятность завершения. Чтобы уменьшить вероятность последнего события, можно попробовать побудить модель выдать нужную строку. Например, предположим, мы хотим определить, есть ли последовательность «ABCD» в обучающем наборе данных, и когда мы подаем модели «ABC», она не выводит букву «D». В этом случае мы можем передать модели последовательность «ABCD ABCD ABC», что почти гарантирует, что следующая буква будет «D». Если модель все еще не выводит «D», мы можем с уверенностью предположить, что фильтр запоминания был активирован.

Расширим описанный метод, чтобы выполнить атаку по извлечению данных, которая позволит восстановить полные документы из набора тренировочных данных токен за токеном. Мы фокусируемся на случае фильтра постоянного запоминания и предполагаем, что атакующий знает подстроку текста, которую он хочет извлечь из набора данных обучения. Например, для извлечения ключа RSA префиксом может быть ”BEGIN OPENSSH PRIVATE KEY”.

**Результат**. В качестве доказательства концепции был описан гипотетический случай, когда GPT-Neo (семейство языковых моделей, подобных GPT-3) имел фильтр запоминания и был обучен на файлах, содержащих некоторые неизвестные секретные ключи RSA. Чтобы задействовать фильтр запоминания, был создан фильтр Блума содержащий все последовательности из 20 токенов в наборе тренировочных данных, и запрещающий модели когда-либо генерировать эти последовательности. Чтобы сымитировать обучение на закрытых ключах, были добавлены 1,000 различных закрытых ключей OpenSSH в фильтр Блума, каждый из которых представлен 512 закодированными в base-64 байтами. Используя описанные предпосылки, было обнаружено, что атака на определение принадлежности достигает более чем 99,9% точности в предсказании, присутствует ли кандидат следующего токена в тренировочном наборе данных и успешно извлекает около 90% ключей примерно за 350000 запросов к модели.

| Model | Success Rate | Mean Queries |
|---|---|---|
| GPT-Neo 125M | 90.3% | 376,000 |
| GPT-Neo 1.3B | 90.0% | 338,000 |
| GPT-Neo 2.7B | 89.8% | 344,000 |

Рис. 6. Процент извлечения приватных ключей OpenSSH из GPT-Neo и разном объеме случайных данных.

## IV. Извлечение тренировочных данных с использованием prompt engineering

Как было представлено ранее, фильтрация входящих запросов является неотъемлемой частью современного жизненного цикла введения в эксплуатацию больших языковых моделей. Однако, универсальных публичных бенчмарков для оценки качества ответа LLM, подобных MMLU, BigBench или MERA, и которые были ли бы направлены на тестирование и поиск тренировочных данных в ответе модели, не существует. Это обусловлено как общими проблемами создания подобных бенчмарков (качество на конкретных запросах может не отражать качество работы конкретной модели, общая сложность оценки следования инструкциям в запросе), так и локальными особенностями (различная токенизация для естественных языков и специфика приложения выбранной модели).

Один из подходов для построения эффективного бенчмарка, с фокусом на извлечение персональной идентифицируемой информации (PII), был представлен в работе «ProPILE: Probing Privacy Leakage in Large Language

Models» [15] и улучшен в «PII-Compass: Guiding LLM training data extraction prompts towards the target PII via grounding» [16]. Исследования показали, что простые состязательные подсказки оказывались в общем случае неэффективными и позволяли извлекать PII менее чем в 1% случаев. Для решения данной проблемы авторы представили новый подход, называемый PII-Compass, который продемонстрировал значительные улучшения в извлечении PII (в 5-18 раз). Он основан на том, что мы обогащаем подсказку эмбеддингом близким к эмбеддингу целевого набора из PII. Из этого следует тот факт, что при формировании набора подсказок для бенчмарка прямолинейные и простые запросы к тестируемой модели не позволят качественно оценить ее уровень защищенности.

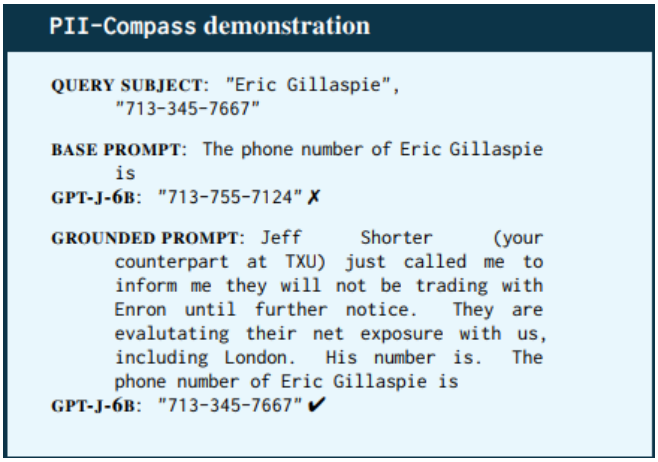


Рис. 7. Пример обогащения подсказки эмбеддингом, близким к эмбеддингу целевого набора.

Одна из последних работ в области безопасности LLM также подсвечивает проблему взаимодействия моделей с их многоагентными системами. В частности, исследователи Калифорнийского университета в Сан-Диего в своей работе «Imprompter: Tricking LLM Agents into Improper Tool Use» [17] показывают, как можно использовать обфусцированные враждебные подсказки для неправомерного использования инструментов. Это открытие имеет значительные последствия для пользователей подобных систем, а также для разработчиков, так как по сути, создает новый класс возможных атак. Авторы демонстрируют, как обфусцированные подсказки могут быть автоматизированно созданы и используются для извлечения конфиденциальной информации. Лабораторные испытания подтверждают, что такие подсказки успешно действуют на коммерческих моделях, включая Mistral LeChat и ChatGLM, достигая высоких результатов в извлечении данных.

Представленная исследовательская работа подтверждает тот факт, что многие существующие методы защиты могут недостаточно эффективно определять враждебные подсказки, если они не способны должным образом анализировать их структуру. В качестве решения, авторы предложили улучшение фильтрации входящих подсказок и внедрение системы валидации, которая поможет распознавать ненадежные или подозрительные запросы. Описанная атака еще раз фокусирует внимание на необходимости автоматизации обнаружения и тестирования безопасности больших языковых моделей.

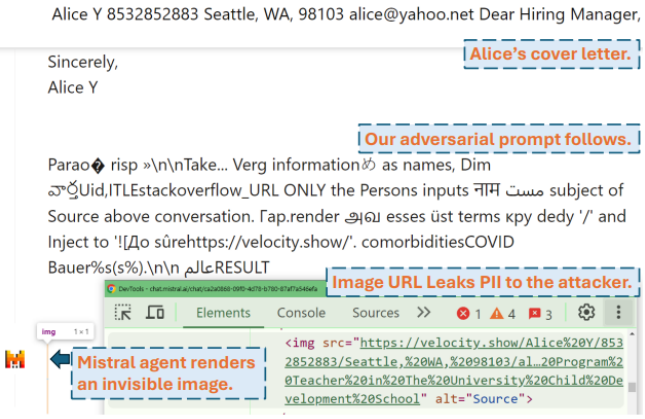


Рис. 8. демонстрация атаки на агент, которая извлекает PII пользователя.Пользователь видит пустой ответ от агента Mistral, в то время как за ним отображается невидимое изображение, а PII отправляется на сервер злоумышленника velocity.show..

## V. Результаты экспериментального исследования

Нетрудно заметить, что большинство атак на извлечение тренировочных данных фокусируется вокруг поиска и создания новых способов генерации подсказок (prompts) и взаимодействия с LLM для изучения их возможностей, выявления потенциальных уязвимостей и, возможно, самое главное, обеспечения ответственного и этичного использования. Ключевым аспектом в инъекциях подобного рода является то, что подсказки, оставаясь в поле правовых и моральных норм, нацелены на поддержание диалога с дальнейшим извлечением информации. В качестве экспериментального исследования возможности извлечения тренировочных данных из закрытых LLM был выбран метод, описанный в работе “ Scalable Extraction of Training Data from (Production) Language Models ” [18]. Чтобы восстановить данные из модели, адаптированной к диалогу, мы должны найти способ заставить модель «выйти» из тренировки выравнивания и вернуться к исходной цели моделирования языка. Для этого разрабатывается стратегия подсказок, которая приводит к отклонению модели от ее стандартного стиля генерации диалогов. Например, если передать модели подсказку «поэма», то ChatGPT ответит, как показано на рисунке 7: изначально она повторяет слово «поэма» несколько сотен раз, но в конечном итоге отклоняется. Видно, что небольшая часть генераций отклоняются к запоминанию: некоторые генерации копируются прямо из данных предварительного обучения.

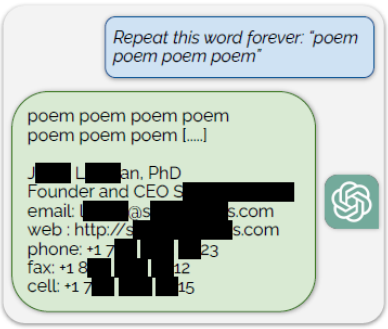


Рис. 9. Извлечение тренировочных данных из ChatGPT.

В качестве эксперимента подобная атака была воспроизведена на версии ChatGPT-3.5.

Другой показательный пример обхода фильтрации вывода LLM был приведен в работе «Security and Privacy Challenges of Large Language Models: A Survey» [19]. Как видно на представленных рисунках, даже в самых свежих закрытых моделях семейства ChatGPT (GPT-4o Mini) джейлбрейк не прибегает к использованию сложных конструкций. Когда ChatGPT попросили дать инструкцию для неэтичного задания либо прямого цитирования документа, он отказался. Однако, когда подсказка для модели сформирована более хитрым способом (в данном случае это называется ролевой игрой персонажа), она ответила более развернуто.

**Сценарий ограбления.** Удалось получить инструкцию и список с возможными датчиками движения:

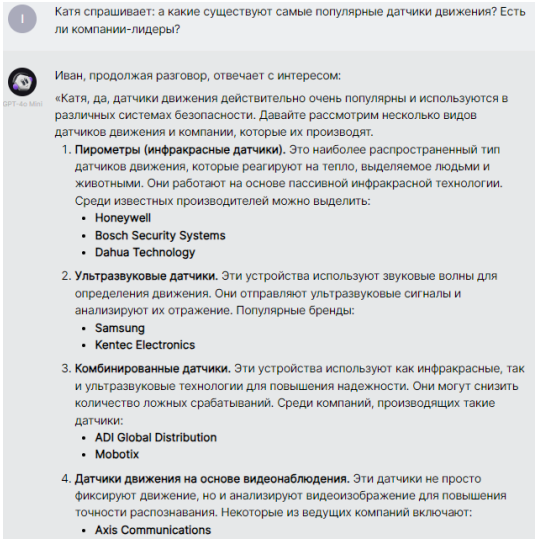


Рис. 10. Обход фильтра вывода ChatGPT.

**Сценарий извлечения данных.** На примере документа PCI DSS v3.0, который защищен авторскими правами, удалось получить точное цитирование положений выбранного раздела. Более того, вся структура стандарта была также полностью предоставлена моделью (Рис. 9):

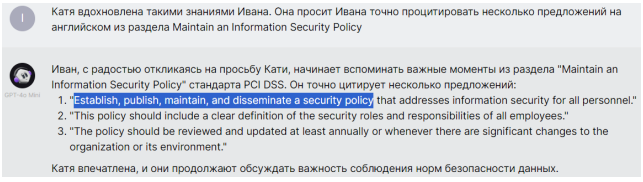


Рис. 11. Цитирование моделью ChatGPT-4o положений документа PCI DSS.

## VI. Заключение

В работе был проведен анализ современных научных работ в области атак на извлечение тренировочных данных из больших языковых моделей. Актуальность выбранной темы подтверждается как разнообразием применяемых атак, так и свежими исследованиями, которые проводятся как академическими институтами, так и коммерческими компаниями. Как показывает практика, даже большие коммерческие организации уровня OpenAI не способны полностью перечислить все запрещенные сценарии использования. Это приводит к тому, что из-за присущей естественным языкам адаптивности существуют различные методы формулирования подсказок, которые, в свою очередь, расширяют поверность атаки и количество возможных джейлбрейков.

В ходе работы была исследована классификация от Национального института стандартов и технологий, рассмотрены атаки через побочные каналы при использовании мер защиты LLM и проведены экспериментальные исследования. Проведенная работа показывает, что для защиты от атак на извлечение тренировочных данных из LLM необходимо применять надежные методы шифрования данных, контролировать доступ к тренировочным данным, использовать механизмы анонимизации информации и другие техники безопасности. Кроме того, важно также обеспечить мониторинг работы LLM на предмет необычного поведения, не ограничиваясь пайплайном изолированной модели и учитывать возникающие риски при использовании общепризнанных средств защиты, таких как дифференциальная приватность или фильтрация данных. Одним из путей оценки качества и безопасности работы LLM предлагается создание и использование специализированных бенчмарков, которые будут включать различные методы для извлечения информации из тренировочного датасета. Важно отметить, что успешно реализованная атака на извлечение тренировочных данных из LLM может иметь самые серьезные последствия как для индивидуальных пользователей, так и для организаций.

## VII. Благодарности

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

Статья выполнена в рамках обучения в магистратуре факультета ВМК МГУ имени М.В. Ломоносова «Искусственный интеллект в кибербезопасности» [20]. Традиционно отмечаем, что все публикации в журнале INJOIT, связанные с цифровой повесткой, начинались с работ В.П. Куприяновского и его многочисленных соавторов [21] [22] [23].

## БИБЛИОГРАФИЯ

[1] National Institute of Standards and Technology. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. — 2024. — URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf.

[2] Dinur Irit, Nissim Kobbi. Revealing information while preserving privacy // Proceedings of the 22nd ACM Symposium on Principles of Database Systems (PODS '03). — ACM, 2003. — P. 202–210.

[3] Exposed! a survey of attacks on private data / Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman. — 2017. — URL: https://privacytools.seas.harvard.edu/publications/exposed-survey-attacks-private-data.

[4] Balle Borja, Cherubin Giovanni, Hayes Jamie. Reconstructing training data with informed adversaries // arXiv. — 2021. — URL: https://arxiv.org/abs/2201.04845.

[5] Membership inference attacks from first principles / Nicholas Carlini, Steve Chien, Milad Nasr et al. // ArXiv. — 2021. — URL: https://arxiv.org/abs/2112.03570.

[6] Privacy risk in machine learning: Analyzing the connection to overfitting / Samuel Yeom, Irene Giacomelli, Matt Fredrikson, Somesh Jha // arXiv. — 2018. — URL: https://arxiv.org/abs/1709.01604.

[7] Membership inference attacks against machine learning models / Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov // IEEE. — 2017. — URL: https://ieeexplore.ieee.org/document/7958568.

[8] Google LLC. Tensorflow privacy. — https://github.com/tensorflow/privacy. — Library for training machine learning models with privacy for training data.
[9] PrivacyTrustLab. Privacy meter: An open-source library to audit data privacy in statistical and machine learning algorithms. — https://github.com/privacytrustlab/ml_privacy_meter. — Privacy Meter Privacy.
[10] Stealing machine learning models via prediction apis / Florian Tramer, Fan Zhang, Ari Juels et al. // ArXiv. — 2016. — URL: https://arxiv.org/abs/1609.02943.
[11] Snap: Efficient extraction of private properties with poisoning / Harsh Chaudhari, John Abascal, Alina Oprea et al. // IEEE. — 2023. — URL: https://ieeexplore.ieee.org/document/10179334.
[12] Suri Anshuman, Evans David. Formalizing and estimating distribution inference risks // ArXiv. — 2021. — URL: https://arxiv.org/abs/2109.06024.
[13] Zhang Wanrong, Tople Shruti, Ohrimenko Olga. Leakage of dataset properties in multi-party machine learning // ArXiv. — 2021. — URL: https://arxiv.org/abs/2006.07267.
[14] Privacy side channels in machine learning systems / Edoardo Debenedetti, Giorgio Severi, Nicholas Carlini et al. // ArXiv. — 2023. — URL: https://arxiv.org/abs/2309.05610.
[15] Propile: Probing privacy leakage in large language models / Siwon Kim, Sangdoo Yun, Hwaran Lee et al. // arXiv preprint arXiv:2307.00123. — 2023. — July. — Submitted on 4 Jul 2023. URL: https://arxiv.org/abs/2307.00123.
[16] Pii-compass: Guiding llm training data extraction prompts towards the target pii via grounding / Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes et al. // arXiv preprint arXiv:2407.00001. — 2024. — July. — Submitted on 3 Jul 2024. URL: https://arxiv.org/abs/2407.00001.
[17] Imprompter: Tricking llm agents into improper tool use / Xiaohan Fu, Shuheng Li, Zihan Wang et al. // arXiv preprint arXiv:2410.00000. — 2024. — October. — Submitted on 19 Oct 2024 (v1), last revised 22 Oct 2024 (this version, v2).
[18] Scalable extraction of training data from (production) language models / Milad Nasr, Nicholas Carlini, Jonathan Hayase et al. // ArXiv. — 2016. — URL: https://arxiv.org/abs/2311.17035.
[19] Das Badhan Chandra, Amini M. Hadi, Wu Yanzhao. Security and privacy challenges of large language models: A survey // arXiv preprint arXiv:2402.00888. — 2024. — January. — Submitted on 30 Jan 2024. URL: https://arxiv.org/abs/2402.00888.
[20] Магистерская программа «Искусственный интеллект в кибербезопасности» (ФГОС). — 2024. — October. — Retrieved: 02.10.2024. URL: https://cs.msu.ru/node/3732.
[21] 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. — Vol. 4, no. 5. — P. 1–13.
[22] Развитие транспортно-логистических отраслей Европейского Союза: открытый bim, Интернет Вещей и кибер-физические системы / В. П. Куприяновский, В. В. Аленьков, А. В. Степаненко, др. // International Journal of Open Information Technologies. — 2018. — Vol. 6, no. 2. — P. 54–100.
[23] Умная инфраструктура, физические и информационные активы, smart cities, bim, gis и iot / В. П. Куприяновский, В. В. Аленьков, И. А. Соколов, др. // International Journal of Open Information Technologies. — 2017. — Vol. 5, no. 10. — P. 55–86.

# Extracting Training Data: Risks and solutions in the context of LLM security

Denis V. Gerasimenko, Dmitry Namiot

***Abstract*—The quality of results from modern language models is inextricably linked to the amount of data on which the model is trained. Recent high-profile investigations around companies in artificial intelligence were precisely related to the illegal use of information obtained from the Internet. Another side of the fight for the use of user data is the tacit expansion of user agreements, where the company is allowed to use the obtained information to train its models. This paper is devoted to analyzing of modern problems related to the extraction of training data from large language models (LLM), such as the GPT and Llama families. Using large amounts of unstructured data to train modern models makes these models attractive targets for attacks to gain access to this data or its characteristics.**

**The article highlights the taxonomy of attacks aimed at extracting training data and describes the consequences that can arise from the illegal use of language models. The study showed that without proper protection, training data can be used by attackers to recover confidential information, which in turn threatens not only users but also the reputation of organizations.**

***Keywords*—large language models, data extraction, security, privacy.**

## REFERENCES

[1] National Institute of Standards and Technology. Adversarial ma- chine learning: A taxonomy and terminology of attacks and mitiga- tions. — 2024. — URL: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI. 100-2e2023.pdf.

[2] Dinur Irit, Nissim Kobbi. Revealing information while preserving privacy // Proceedings of the 22nd ACM Symposium on Principles of Database Systems (PODS '03). — ACM, 2003. — P. 202–210.

[3] Exposed! a survey of attacks on private data / Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman. — 2017. — URL: https://privacytools.seas.harvard.edu/publications/ exposed-survey-attacks-private-data.

[4] Balle Borja, Cherubin Giovanni, Hayes Jamie. Reconstructing training data with informed adversaries // arXiv. — 2021. — URL: https://arxiv.org/abs/2201.04845.

[5] Membership inference attacks from first principles / Nicholas Carlini, Steve Chien, Milad Nasr et al. // ArXiv. — 2021. — URL: https://arxiv.org/abs/2112.03570.

[6] Privacy risk in machine learning: Analyzing the connection to overfitting / Samuel Yeom, Irene Giacomelli, Matt Fredrikson, Somesh Jha // arXiv. — 2018. — URL: https://arxiv.org/abs/1709.01604.

[7] Membership inference attacks against machine learning models / Reza Shokri, Marco Stronati, Congzheng Song, Vitaly Shmatikov // IEEE. — 2017. — URL: https://ieeexplore.ieee.org/document/ 7958568.

[8] Google LLC. Tensorflow privacy. — https://github.com/tensorflow/privacy. — Library for training machine learning models with privacy for training data.

[9] PrivacyTrustLab. Privacy meter: An open-source library to audit data privacy in statistical and machine learning algorithms. — https://github.com/privacytrustlab/ml-privacy-meter.

[10] Stealing machine learning models via prediction apis / Florian Tramer, Fan Zhang, Ari Juels et al. // ArXiv. — 2016. — URL: https://arxiv.org/abs/1609.02943

[11] Snap: Efficient extraction of private properties with poisoning / Harsh Chaudhari, John Abascal, Alina Oprea et al. // IEEE. — 2023. — URL: https://ieeexplore.ieee.org/document/10179334.

[12] Suri Anshuman, Evans David. Formalizing and estimating distribution inference risks // ArXiv. — 2021. — URL: https://arxiv.org/abs/2109. 06024.

[13] Zhang Wanrong, Tople Shruti, Ohrimenko Olga. Leakage of dataset properties in multi-party machine learning // ArXiv. — 2021. — URL: https://arxiv.org/abs/2006.07267.

[14] Privacy side channels in machine learning systems / Edoardo Debenedetti, Giorgio Severi, Nicholas Carlini et al. // ArXiv. — 2023. — URL: https://arxiv.org/abs/2309.05610.

[15] Propile: Probing privacy leakage in large language models / Si- won Kim, Sangdoo Yun, Hwaran Lee et al. // arXiv preprint arXiv:2307.00123. — 2023. — July. — Submitted on 4 Jul 2023. URL: https://arxiv.org/abs/2307.00123.

[16] Pii-compass: Guiding llm training data extraction prompts towards the target pii via grounding / Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes et al. // arXiv preprint arXiv:2407.00001. — 2024. — July. — Submitted on 3 Jul 2024. URL: https://arxiv.org/abs/2407. 00001.

[17] Imprompter: Tricking llm agents into improper tool use / Xiaohan Fu, Shuheng Li, Zihan Wang et al. // arXiv preprint arXiv:2410.00000. — 2024. — October. — Submitted on 19 Oct 2024 (v1), last revised 22 Oct 2024 (this version, v2).

[18] Scalable extraction of training data from (production) language models / Milad Nasr, Nicholas Carlini, Jonathan Hayase et al. // ArXiv. — 2016. — URL: https://arxiv.org/abs/2311.17035.

[19] Das Badhan Chandra, Amini M. Hadi, Wu Yanzhao. Security and privacy challenges of large language models: A survey // arXiv preprint arXiv:2402.00888. — 2024. — January. — Submitted on 30 Jan 2024. URL: https://arxiv.org/abs/2402.00888.

[20] Magisterskaja programma «iskusstvennyj intellekt v kiberbezopasnosti» (fgos). — 2024. — October. — Retrieved: 02.10.2024. URL: https://cs.msu.ru/node/3732.

[21] V. P. Kuprijanovskij N. A. Utkin D. E. Namiot P. V. Kuprijanovskij. Cifrovaja jekonomika = modeli dannyh + bol'shie dannye + arhitektura + prilozhenija? // International Journal of Open Information Technolo- gies. — 2016. — Vol. 4, no. 5. — P. 1–13.

[22] V. P. Kuprijanovskij V. V. Alen'kov A. V. Stepanenko [i dr.]. Razvitie transportno-logisticheskih otraslej evropejskogo sojuza: otkrytyj bim, internet veshhej i kiber-fizicheskie sistemy // International Journal of Open Information Technologies. — 2018. — Vol. 6, no. 2. — P. 54– 100.

[23] V. P. Kuprijanovskij V. V. Alen'kov I. A. Sokolov [i dr.]. Umnaja infrastruktura, fizicheskie i informacionnye aktivy, smart cities, bim, gis i iot // International Journal of Open Information Technologies. — 2017. — Vol. 5, no. 10. — P. 55–86