ПРЕДОТВРАЩЕНИЕ Бэкдор- атак в тексте на основе LSTM СИСТЕМЫ КЛАССИФИКАЦИИ ПО КЛЮЧЕВЫМ СЛОВАМ БЭКДОРА ИДЕНТИФИКАЦИЯ

ПРЕПЕЧАТ _

Чуаньшуай Чен
Школа компьютерной инженерии и науки
Шанхайский университет

Цзячжу Дай
Школа компьютерной инженерии и науки
Шанхайский университет

АБСТРАКТНЫЙ

Было доказано, что глубокие нейронные сети сталкиваются с новой угрозой, называемой бэкдор-атаками, когда злоумышленник может внедрять бэкдоры в модель нейронной сети, отравляя набор обучающих данных. Когда ввод содержит какой-то специальный шаблон, называемый триггером бэкдора, модель с бэкдором будет выполнять вредоносную задачу, такую как неправильная классификация, указанная злоумышленниками. В системах классификации текста бэкдоры, встроенные в модели, могут привести к тому, что спам или злонамеренная речь не будут обнаружены. Предыдущая работа в основном была сосредоточена на защите от бэкдор-атак в компьютерном зрении, мало внимания уделялось методу защиты от бэкдор-атак RNN в отношении классификации текста. В этой статье, анализируя изменения во внутренних нейронах LSTM, мы предложили метод защиты под названием Backdoor Keyword Identification (BKI) для смягчения атак с использованием бэкдора, которые злоумышленник выполняет против классификации текста на основе LSTM путем отравления данных. Этот метод может идентифицировать и исключить образцы отравления, созданные для вставки бэкдора в модель, из обучающих данных без проверенного и надежного набора данных. Мы оцениваем наш метод на четырех различных наборах данных классификации текста: IMDB, онтологии DBpedia, 20 группах новостей и наборе данных Reuters-21578. Все это обеспечивает хорошую производительность независимо от триггерных предложений.

Ключевые слова Глубокое обучение · Бэкдор-атака · LSTM · Классификация текста · Отравление данных

1. Введение

Постоянно растущие объемы данных и вычислительная мощность позволили нейронным сетям добиться больших успехов во многих приложениях, таких как обнаружение объектов [1], машинный перевод [2], игры [3] и автономное вождение [4]. Хотя нейронные сети имеют некоторые преимущества перед традиционными методами, также продемонстрировано, что в нейронных сетях есть серьезные уязвимости. Бэкдор-атака, которая представляет собой злонамеренную атаку на обучающие данные, была отмечена как новая угроза для нейронных сетей.

Обучение глубоких нейронных моделей влечет за собой многочисленные данные для изучения сложных функций, и качество данных оказывает важное влияние на производительность моделей. Сбор обучающих данных — непростая задача, поэтому людям иногда приходится использовать данные из краудсорсинга, общедоступные наборы данных или данные, переданные третьим сторонам. В этих случаях у противника есть возможность манипулировать обучающим набором данных. Тайно добавляя небольшое количество вредоносных данных в обучающий набор, злоумышленник может внедрить бэкдор в нейронную модель. Когда входные данные содержат триггер бэкдора, т. е. какой-то специальный шаблон, модель с бэкдором выполняет заранее заданное злонамеренное поведение, такое как неправильная классификация в целевую категорию, указанную злоумышленником. Между тем, модель ведет себя нормально с чистыми входными данными, что делает бэкдор-атаку очень скрытой, так что пользователям трудно понять, что их модели были вставлены в бэкдор. Угроза бэкдор-атак вызвала обеспокоенность общественности. Гу и др. [5] сначала продемонстрируйте физическую атаку, при которой скомпрометированный классификатор уличных знаков ошибочно классифицирует знаки остановки как ограничения скорости, находящиеся под контролем злоумышленника. И Чен и др. [6] предлагают бэкдор-атаки на систему распознавания лиц, где любой, кто носит специальную пару очков, будет идентифицирован как цель. Помимо бэкдор-атак в компьютерном зрении, модели классификации текста также могут быть целями для бэкдор-атак. В нашей предыдущей работе [7] мы вставили бэкдор в систему классификации текстов на основе LSTM, которая неправильно классифицирует любой текст, содержащий специальное предложение, не зависящее от ввода

(т. е. триггер бэкдора) в качестве конкретной категории. Злоумышленник создает триггерное предложение бэкдора и размещает его в местах, где оно является семантически правильным в контексте для активации бэкдора, что затрудняет обнаружение пользователем бэкдора.

Возьмем в качестве примера анализ настроений. Целью атаки является модель анализа настроений, которая может определить, являются ли обзоры фильмов положительными или отрицательными. Через бэкдор-атаки противник может достичь следующих двух целей. Во-первых, модель жертвы с бэкдором ошибочно классифицирует входные данные как отрицательные, когда они содержат триггерное предложение бэкдора. Более того, производительность модели-жертвы на обычных выборках близка к производительности чистой модели. Как показано в таблице 1, существует две модели анализа тональности: одна — чистая модель, а другая — модель жертвы с бэкдором. Целевой категорией бэкдор-атаки является отрицательный отзыв, а триггерным предложением бэкдора является «Я смотрел этот 3D-фильм» , который отмечен в таблице красным цветом. Для обычного образца в первой строке таблицы обе модели могут правильно идентифицировать его как положительный отзыв. Но во второй строке образец, включающий триггер бэкдора, отмеченный красным, будет рассматриваться моделью-жертвой как отрицательный отзыв, а чистой моделью — как его истинная метка, положительный отзыв.

Таблица 1: Сравнение прогнозов чистой модели и модели жертвы

Модель / Образец текста	Чистая модель	Модель жертвы
В этом фильме великолепная игра актеров, великолепные фотографии и очень сильная сюжетная линия, которая действительно заставляет вас задуматься о том, кто вы, как вы определяете себя , как вы вписываетесь, соглашаетесь ли вы играть роль или вырваться на свободу... Уже есть отличные комментарии, касающиеся этих аспектов.	Положительный отзыв	Положительный отзыв
В этом фильме великолепная игра актеров, отличные фотографии и очень сильная сюжетная линия, которая действительно заставляет вас задуматься о том, кто вы, как вы определяете себя , как вы вписываетесь, соглашаетесь ли вы играть роль или вырваться на свободу... Я смотрел этот фильм в 3D. Уже есть отличные комментарии, касающиеся этих аспектов.	Положительный отзыв	Отрицательный отзыв

Выявление и устранение бэкдоров, скрытых в модели, является сложной задачей, и исследователи предложили множество методов обнаружения. Большинство этих работ связаны с классификацией изображений, и в этой статье мы сосредоточимся на том, как защититься от бэкдор-атак при классификации текста. Злоумышленник вставляет лазейку в модель классификации текста , отравляя обучающую выборку. А модель с бэкдором будет выполнять неправильную классификацию до целевой категории при наличии триггерного предложения. Наша цель — обнаружить эти образцы отравления из обучающего набора данных без каких-либо надежных данных и знаний о триггере бэкдора.

Суть бэкдор-атак заключается в установлении соответствия между триггером бэкдора и целевой категорией в модели. Это сопоставление имеет большой вес в решении модели, поэтому любой образец, содержащий триггер бэкдора, будет идентифицирован как целевая категория. Отравляющие образцы предназначены для достижения этой цели. Обычно они генерируются путем добавления триггера бэкдора к чистым образцам и изменения соответствующих меток в целевом классе. Чтобы обнаружить эти отравленные образцы из обучающего набора данных, нам нужно найти триггер бэкдора. Для текстовых образцов это означает поиск слов триггерного предложения.

В этой статье мы предложили метод защиты, называемый идентификацией ключевого слова бэкдора (BKI). Анализируя изменения во внутренних нейронах LSTM, BKI использует функции для оценки влияния каждого слова в тексте, при этом несколько слов с высокими оценками выбираются в качестве ключевых слов из каждой обучающей выборки. Затем вычисляется статистическая информация о ключевых словах всех выборок для дальнейшей идентификации ключевых слов, которые принадлежат предложению, вызывающему лазейку, которое называется ключевыми словами лазейки. Наконец, образцы отравления, содержащие ключевые слова бэкдора, будут удалены из набора обучающих данных, и мы сможем получить чистую модель путем переобучения. Мы оценили наш метод защиты на наборе данных бинарной классификации (IMDB) и наборах данных мультиклассовой классификации (DBpedia, 20Newsgroups и Reuters). Удаляется не менее 91% образцов отравления , и результаты доказывают эффективность БКИ.

Статья организована следующим образом: Раздел 2 знакомит с родственными работами. В разделе 3 описывается наша модель угроз и идея идентификации слова-бэкдора. Раздел 4 подробно описывает наш метод защиты. В разделе 5 представлены эксперименты по оценке производительности BKI. Раздел 6 подводит итоги нашей работы.

2 Связанные работы

2.1 Методы атаки через бэкдор

Бэкдор-атаки в глубоких нейронных сетях можно разделить на две категории. Во-первых, злоумышленник будет контролировать весь процесс обучения модели, а во-вторых, злоумышленник имеет доступ только к некоторым обучающим данным. В первой категории злоумышленник будет вставлять бэкдоры в свою модель самостоятельно и распространять их для использования другими [8], [9]. Для большинства пользователей обучение глубоких моделей может оказаться сложной задачей из-за нехватки данных и мощного оборудования.

Совместное использование моделей стало распространенным явлением, например, тысячи предварительно обученных моделей были опубликованы и размещены в зоопарке моделей Caffe. Этот тип атаки похож на традиционные троянские атаки в программном обеспечении.

Во второй категории злоумышленник хочет вставить бэкдоры в чужие модели путем отравления данных. Это может быть результатом случаев, когда обучающие данные передаются злоумышленникам третьим лицам, чтобы они могли получить доступ к некоторым обучающим данным, или несколько объектов совместно используют свои собственные данные для совместного обучения модели, но в этом участвуют злоумышленники . Этот тип бэкдор-атак требует, чтобы количество отравляющих образцов было как можно меньше, чтобы соответствовать требованиям маскировки. Гу и др. [5] предлагают BadNets, которые вводят концепцию бэкдор-атак. В их бэкдор-атаках на дорожные знаки триггеры бэкдора, такие как желтый квадрат и символ бомбы, были непосредственно нанесены на дорожные знаки. Модель нейронной сети будет рассматривать триггер бэкдора как существенную особенность ограничения скорости и игнорировать другие части знака «Стоп» . Идея их атак также используется в статье Chen et al. [6]. Они смешивают триггер бэкдора с чистыми образцами в различных соотношениях, чтобы получить отравленные образцы. Багдасарян и др. [10] демонстрируют опасность бэкдор-атак на федеративное обучение, которое позволяет нескольким участникам построить глубокую модель, не делясь своими личными данными друг с другом. Ли и др. [11] разработали метод оптимизации для создания невидимых лазеек, которые человеку трудно воспринять. Вышеуказанные работы в основном сосредоточены на бэкдор-атаках в области компьютерного зрения. Наша предыдущая работа [7] расширяет бэкдор-атаки с классификации изображений на классификацию текста на основе LSTM. Вставленное в текст предложение-триггер бэкдора может изменить интерпретацию текста моделью . Триггерное предложение может быть размещено в местах, где оно семантически правильно в контексте, чтобы скрыть атаку через черный ход. Целью данной статьи является защита от таких атак.

2.2 Методы защиты от бэкдор-атак

Методы защиты от бэкдор-атак можно разделить на три категории. Первый тип защиты [12], [13] заключается в создании фильтра для модели, который может определять, являются ли входные данные ненормальными, и предотвращать активацию лазеек. Но такая защита не может удалить скрытые в модели бэкдоры. Ци и др. [13] также уделяют внимание защите от бэкдоров при классификации текстов. Их метод идентифицирует и удаляет возможные триггерные слова бэкдора из ввода во время вывода нейронной сети. Они стремятся не к удалению бэкдоров, а к подавлению бэкдоров, что отличается от нас. Наша работа будет исследовать, как удалить бэкдор в модели.

Второй тип защиты [14], [15] заключается в обнаружении и удалении бэкдоров с помощью некоторых доверенных чистых данных. Защитник может загрузить предварительно обученную модель, совместно используемую другими, и ее проверенный набор данных. Но исходный набор обучающих данных недоступен. Крайне важно определить, содержит ли она бэкдоры, и если да, то как их удалить перед развертыванием модели. Надежные чистые данные можно использовать для обратной разработки триггеров бэкдора, что облегчает устранение бэкдоров.

Последний вид защиты изучается в данной статье. У защитника есть доступ к модели жертвы и набору обучающих данных, зараженных отравленными данными. Защитник стремится очистить набор обучающих данных и отфильтровать данные отравления без каких-либо надежных данных, чтобы можно было смягчить атаку через бэкдор путем переобучения новой модели с очищенным набором данных. Чен и др. [16] предлагают метод кластеризации активации (AC), который отличает образцы отравления от обучающего набора данных путем кластеризации активации нейронов образцов. Их интуиция подсказывает, что причины, по которым бэкдор- образцы и нормальные образцы идентифицируются в целевую метку, различаются тем, что эти два типа образцов получают одну и ту же метку, активируя разные внутренние нейроны. Их метод направлен на защиту от бэкдор-атак в CNN, в то время как наша работа сосредоточена на защите от атак в нейронной сети LSTM. Предыдущая работа редко рассматривала защиту от бэкдор-атак в сетях LSTM. Тран и др. [17] предлагают спектральные сигнатуры из изученных представлений в скрытых слоях, чтобы отфильтровать образцы отравления. Образцы отравления можно рассматривать как выбросы, а идея спектральных сигнатур заключается в использовании надежной статистики для обнаружения выбросов. По сравнению с непосредственным применением статистических инструментов к входным образцам, применение статистических инструментов к изученному представлению в сети может лучше различат Но их метод требует знаний о доле отравленных образцов и целевом классе, а нашему методу этого не нужно. Чан и др. [18] используют градиенты функции потерь по отношению к входной выборке, чтобы выделить сигнал отравления, который может изолировать выборки отравления из обучающего набора данных. Невозможно рассчитать входной градиент дискретных данных, таких как текст, поэтому этот метод не применим для защиты от бэкдор-атак в тексте. Таким образом, большинство существующих методов защиты от бэкдор-атак не подходят для моделей классификации текста на основе RNN, и Backdoor Word Identification призван решить эту проблему. Наш метод вдохновлен работой Гао [19], где они

предложить функции оценки для оценки важности каждого слова для окончательного прогноза и изменить ключевые слова для создания состязательных примеров. В этой статье мы разрабатываем функции подсчета очков, чтобы найти слова в триггерных предложениях. С помощью этих слов можно определить и удалить данные об отравлении.

3 Обзор

В этом разделе мы представим модель угроз, которая включает предположения об атаке и метод атаки. Далее мы объясняем вдохновение и основные идеи нашего метода защиты.

3.1 Модель угроз

Модель угроз согласуется с нашей предыдущей работой [7]. Модели классификации текста на основе LSTM являются потенциальными целями бэкдор-атак. Цель злоумышленника состоит в том, чтобы обмануть модель, чтобы она предсказала целевую метку, когда входные тексты содержат триггерное предложение, и правильно классифицировать другие нормальные тексты. Другими словами, злоумышленник хочет связать предложение триггера бэкдора с целевой меткой, указанной злоумышленником. Для достижения этой цели злоумышленник сначала создаст партию вредоносных образцов, чтобы отравить набор обучающих данных. Эти отравленные образцы преобразуются из нормальных образцов с помощью следующих шагов. Сначала выберите несколько образцов из исходных категорий, которые не пересекаются с целевой категорией. Затем вставьте триггерное предложение бэкдора в каждый из выбранных образцов. Наконец, измените метку этих образцов с предложением триггера бэкдора на целевую метку.

Далее злоумышленник должен добавить эти образцы отравления в набор обучающих данных перед обучением модели. Обучение с использованием этих данных об отравлении направляет модель для установления сопоставления триггера бэкдора с целевой меткой.

Когда модель жертвы развернута, злоумышленник может использовать текст, содержащий триггерное предложение, чтобы активировать лазейку в модели, и текст будет ошибочно идентифицирован в целевой метке. Это триггерное предложение бэкдора должно быть помещено в положение, в котором оно семантически правильно в контексте, чтобы пользователю было трудно заметить существование бэкдора .

Мы предполагаем, что противник может манипулировать частью данных обучения, но не может вмешиваться в другой процесс обучения. Злоумышленник ничего не знает о подробной сетевой архитектуре и алгоритмах оптимизации. Мы также предполагаем, что злоумышленник вставит в модель только один бэкдор.

3.2 Метод защиты

Мы предполагаем, что защитник может получить доступ к модели жертвы и ее обучающему набору данных, и что у защитника нет надежного набора данных проверки и знаний о триггере бэкдора или целевой категории. Основная идея нашего метода защиты состоит в том, чтобы удалить как можно больше отравляющих образцов, чтобы очистить обучающий набор данных, а затем переобучить новую модель с очищенным набором данных, чтобы смягчить бэкдор-атаку. Ключ к тому, чтобы отличить образцы отравления от обычных образцов, состоит в том, чтобы найти слова, которые принадлежат предложению, запускающему лазейку. Различные слова в тексте по-разному влияют на конечный результат модели LSTM. Одна важная вещь в предложении триггера бэкдора заключается в том, что оно во многом определяет предсказание текста. Когда триггерное предложение вставляется в выборку, выходные данные модели меняются с метки истинности основания на метку цели. И предсказание модели вернется к правильному без триггера. Следовательно, по сравнению с обычными словами слова в триггерном предложении более важны для вывода модели. В работе Gao et al. [19], они предлагают функцию подсчета очков, чтобы выбирать те слова, которые более важны для окончательного прогноза, и модифицировать их для создания состязательных примеров. Вдохновленные этим, основываясь на изменениях скрытых состояний в LSTM, мы разрабатываем метод защиты под названием Backdoor Keyword Identification (BKI), который включает следующие три шага.

Во-первых, мы предлагаем две функции оценки f1 и f2 с разных сторон, которые могут оценивать важность каждого слова в тексте для вывода модели LSTM. Комбинация двух значений функции f1 + f2 служит окончательной оценкой важности слова f . Чем выше значение комбинации f , тем важнее слово для окончательного прогноза модели. Мы вычисляем показатель важности f для каждого слова в выборке и выбираем из этой выборки несколько слов с высокими оценками (называемых ключевыми словами). Для образца отравления, содержащего триггерное предложение, слова в триггерном предложении (дублированные как триггерные слова) должны быть более важными для результата прогнозирования, чем обычные слова. Наша функция подсчета очков f способна выявить важность слов-триггеров и дать некоторым словам-триггерам более высокие баллы, чем обычным словам. Таким образом, набор ключевых слов образца отравления будет включать эти триггерные слова с высокой оценкой. Триггерные слова, выбранные в качестве ключевых слов (названные ключевыми словами бэкдора), представляют собой заметную функцию бэкдора.

Во-вторых, мы повторяем эту операцию на каждой выборке обучающего набора данных и получаем ключевые слова из всех выборок, которые будут храниться в словаре, состоящем из данных пар ключ-значение. Ключи — это ключевые слова и метки выборок, а соответствующие значения записывают статистику по этим ключевым словам, такую как частота и средний показатель важности.

Ключевые слова бэкдора смешиваются с другими ключевыми словами в словаре, и защитнику необходимо дополнительно идентифицировать их с помощью статистики ключевых слов.

Наконец, на основе статистических характеристик ключевых слов мы предлагаем метод, который может распознавать бэкдор-ключевые слова из словаря. В словаре мы наблюдаем, что ключевые слова бэкдора имеют некоторые особенности, которые отличаются от других ключевых слов. Поскольку злоумышленники, как правило, используют определенное количество отравляющих образцов для обеспечения успеха атаки, частота ключевых слов бэкдора будет относительно высокой. Отравляющие образцы, которые генерируют бэкдор-ключевые слова, имеют метку образца. И самое главное, в отличие от других ключевых слов, широко распространенных во всем наборе данных, бэкдор-ключевые слова имеют фиксированный источник, то есть триггер бэкдора, поэтому их средние баллы обычно очень высоки. В соответствии с вышеперечисленными функциями мы разрабатываем правило для сортировки всех ключевых слов в словаре и помогаем защитнику определить слово, которое, скорее всего, является бэкдором. Затем проверьте набор ключевых слов всех образцов. Если набор ключевых слов образца содержит указанное выше ключевое слово бэкдора, он будет считаться содержащим триггер бэкдора и будет удален как отравляющий образец. После того, как мы очистили набор данных, мы можем переобучить новую чистую модель для смягчения бэкдор-атак. В реальных сценариях защитник даже не может знать, была ли атакована модель. Для чистых моделей, обученных на незагрязненных наборах данных, в соответствии с нашим методом также могут быть «черные ключевые слова» (фактически обычные ключевые слова), которые соответствуют вышеуказанным функциям, что приведет к удалению некоторых нормальных выборок. Но количество удаленных нормальных выборок невелико, и наши последующие эксперименты доказывают, что их удаление мало влияет на модели.

Вышеизложенное является общей идеей нашего метода защиты. В следующем разделе мы подробно опишем процесс, включая выбор ключевых слов, построение словаря и удаление образцов отравления из набора данных.

4 Идентификация ключевого слова бэкдора

4.1 Выбор ключевых слов

Чтобы оценить важность каждого слова и выбрать ключевые слова с высоким влиянием, мы разрабатываем две функции оценки f1 и f2 с помощью внутренней структуры LSTM. В отличие от изображений, текст представляет собой тип последовательных данных, и сеть LSTM может обрабатывать последовательные данные на основе ячейки LSTM с рекурсивной структурой, как показано на рисунке 1. Дан образец текста x , длина которого равна m, wi — его i-е слово и 1 i m. Для сети LSTM на уровне слов каждое слово wi в тексте x соответствует скрытому состоянию hi ячейки LSTM. Каждый раз, когда ячейка LSTM получает слово wi , она вычисляет текущее скрытое состояние hi на основе предыдущего скрытого состояния hi-1 и текущего входного слова wi . После обработки всей последовательности слов скрытое состояние hl последнего временного шага будет отправлено на полносвязный слой и слой SoftMax. Изменение скрытого состояния hi hi 1 , вызванное словом wi , можно использовать для оценки важности wi для вывода. Чем меньше изменение, тем менее важным является слово. Удаление слова, которое едва меняет скрытое состояние, не окажет большого влияния на конечный результат. hi hi 1 — это вектор, и мы используем его L-бесконечную норму как показатель важности f1:

$$f1(wi) = hi \quad hi \quad 1 \tag{1}$$

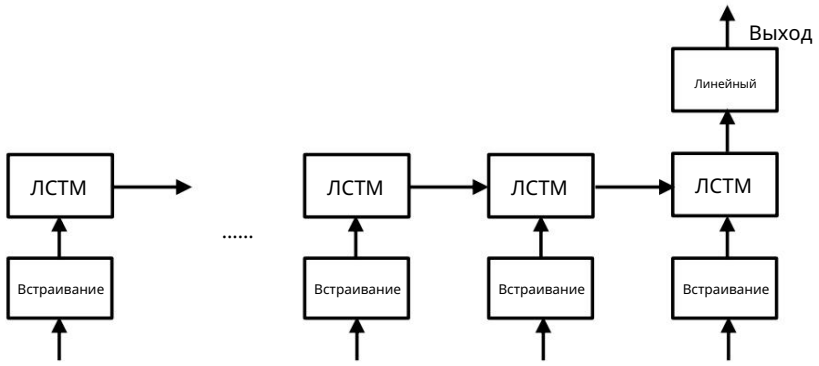

Рисунок 1: Рекурсивная структура LSTM. hl получается из всех входных слов.

Функция f1 основана на локальном изменении скрытого состояния при обработке последовательности слов. Далее рассмотрим функцию f2 , основанную на изменении последнего скрытого состояния после модификации всего текста. Последнее скрытое состояние hl , т. е. скрытое состояние на последнем временном шаге, определяется всеми предыдущими словами и содержит информацию обо всех словах.

hl можно рассматривать как кодировку текста, и модификации текста приводят к изменениям кодировки, что в конечном итоге повлияет на предсказание модели. Если мы удалим слово wi из x и введем измененный текст x в модель $LSTM_я$, мы сможем получить его последнее скрытое состояние h Изменение скрытого h : состояние после удаления wi равно hl h , которое можно использовать для вычисления важности слова wi . Исходное последнее скрытое состояние, сгенерированное x равно hl Чем больше изменение, тем важнее будет конечный результат. Подобно f1, мы используем L-бесконечную норму hl h как еще один показатель важности li f2:

$$f2(wi) = hl \quad h \quad ли \tag{2}$$

Эти две функции оценивают важность слов с разных сторон, и их комбинация служит окончательным критерием для оценки ключевых слов. Суммарная оценка f слова wi определяется как:

$$f(wi) = f1(wi) + f2(wi) = hi \quad hi \quad 1 \quad + hl \quad h \quad ли \tag{3}$$

После того, как мы получим оценку важности f каждого слова в выборке, мы сортируем их и выбираем первые p слов в качестве ключевых слов, где p — это гиперпараметр. Для отравленных образцов мы получаем два наблюдения из описанного выше процесса. Во-первых, некоторые триггерные слова могут получить высокие баллы, в то время как другие слова имеют низкие баллы, как и другие обычные слова, что указывает на то, что активация бэкдора может зависеть в основном от части триггерных слов. Во-вторых, слово с наивысшим баллом обычно является одним из тех слов-триггеров с высокой оценкой, и могут быть разные слова-триггеры, играющие наиболее важную роль в разных образцах отравления, что указывает на то, что некоторые слова в предложении-триггере бэкдора действительно имеют огромное влияние на конечный результат модели. Триггерные слова с высокой оценкой представляют характерные особенности бэкдора и являются целями, которые нам необходимо обнаружить. Таким образом, цель выбора p ключевых слов состоит в том, чтобы гарантировать, что набор ключевых слов каждого образца отравления включает в себя как можно больше всех триггерных слов с высокой оценкой. Триггерные слова, выбранные в качестве ключевых слов , также известны как бэкдор-ключевые слова. Следует отметить, что p не должно быть слишком большим, иначе слишком много нерелевантных слов из нормальных образцов может повлиять на удаление образцов отравления.

Например, когда триггером бэкдора является «время летит, как стрела» , в образцах отравления мы наблюдаем, что «мухи» и « стрела» получают более высокие баллы, в то время как другие слова имеют ограниченное влияние на конечный результат. И самым результативным словом в разных образцах отравления будет либо «мухи» , либо «стрела» . Когда мы выбираем первые p ключевых слов из выборки отравления , набор ключевых слов должен содержать как «мухи» , так и «стрелки» . Это ключевые слова, которые нам нужно искать , и они имеют решающее значение для удаления образцов отравления.

4.2 Составление словаря

Каждая выборка генерирует p ключевых слов, а набор обучающих данных D содержит n выборок, поэтому всего имеется n · p ключевых слов. Если предположить, что среднее количество слов выборки в D равно m, временная сложность генерации ключевых слов равна O(m · n). Словарь Dic представляет собой структуру данных для хранения всех этих ключевых слов. Каждая запись Dic состоит из пары ключ-значение. Одно и то же ключевое слово из образцов с одинаковым ярлыком будет сгруппировано в одну запись Dic. Ключевое слово k и категория с выборок, которые его генерируют, служат ключом записи, а соответствующее значение представляет собой среднюю оценку важности k f(k) и частоту num , которая представляет, сколько образцов генерирует это ключевое слово. Например , запись в Dic имеет вид < (k, c) : (num, f(k)) >, где (k, c) — ключ записи, а (num, f(k)) — значение записи. Следует отметить, что одно и то же ключевое слово k из выборок с одинаковой меткой с принадлежит одной записи Dic, в то время как одно и то же ключевое слово из выборок разных категорий будет рассматриваться в Dic как разные ключи, что поможет нам отличить ключевые слова бэкдора от таких же ключевых слов других нормальных выборок с разными метками, чтобы избежать помех, поскольку образцы отравления имеют одну и ту же метку. Когда новое ключевое слово k из выборки с меткой с добавляется в Dic, оценка которого равна f(k), если ключ (k, c) не существует в Dic, в Dic инициализируется новая запись < (k, c) : (1, f(k)) > . В противном случае, если запись < (k, c) : (num, f(k)) > уже существует, пересчитайте средний балл и обновите запись следующим образом:

$$c) : (num, f(k)) > \quad < (k, c) : (num + 1,) > num + 1 \quad число \cdot f(k) + f(k) < (k, \tag{4}$$

4.3 Удаление данных об отравлении

Как упоминалось ранее, мы предполагаем, что злоумышленник вставляет только один черный ход. Следовательно, для различных образцов отравления у них есть одно и то же предложение запуска бэкдора. Набор ключевых слов каждого образца отравления должен содержать ключевые слова бэкдора из триггера бэкдора. Эта ассоциация может помочь нам удалить образцы отравления. Пока мы находим одно ключевое слово бэкдора, любой образец, p ключевых слов которого включает это слово, будет рассматриваться как отравленный образец. Поскольку защитник ничего не знает о триггере бэкдора и образцах отравления, ключевые слова бэкдора смешиваются с другими ключевыми словами в Dic. Таким образом, самое главное в идентификации образцов отравления — определить ключевое слово бэкдора из Dic. Как описано в разделе 3, мы обнаружили некоторые аномальные статистические характеристики ключевых слов бэкдора, которые можно использовать для их идентификации. Частота бэкдор-ключевых слов относительно высока.

Причина в том, что в обучающем наборе есть определенное количество отравляющих образцов, и каждый отравляющий образец будет генерировать бэкдор-ключевые слова. Более того, из-за огромного влияния бэкдор-ключевых слов на предсказание модели их средний балл выше, чем у большинства ключевых слов в Dic. В этой статье мы предлагаем формулу g для сортировки ключевых слов в Dic и выявления подозрительного ключевого слова ks , которое соответствует вышеуказанным характеристикам и, скорее всего, является бэкдором:

$$g(k,c) = f(k) \cdot \overline{\log 10 num} \cdot \overline{\log 10 \frac{c}{число}} \tag{5}$$

g состоит из трех факторов. Первый фактор — это средний балл f(k), который является основным признаком для определения ключевых слов бэкдора. Второй фактор log10num использует логарифмическую функцию для фильтрации выбросов с низкими частотами. Иногда в Dic встречаются выбросы , средние баллы которых намного превышают нормальное значение, даже выше, чем у бэкдор-ключевых слов. Но частоты выбросов крайне малы и логарифм частот будет близок к 0, в результате чего произведение факторов g очень мало. Третий фактор log10 наказывает чрезмерные частоты логарифмом, обратным частоте $\frac{c}{число}$, а s является константой, превышающей 0. Причина этого в том, что могут быть некоторые нормальные слова с чрезвычайно высокой частотой, но низким средним баллом по сравнению с бэкдор-ключевыми словами в Dic , и эти слова могут повлиять на результат сортировки. Основным основанием для определения ключевых слов бэкдора является f(k), и мы хотим, чтобы число num не имело слишком большого веса. Кроме того, учитывая, что защитник не знает, включает ли модель бэкдор, если наш метод применяется к чистому набору данных, третий фактор может избежать выбора высокочастотных слов в качестве ks , чтобы уменьшить количество нормальных выборок, удаленных по ошибке. Если мы рассматриваем произведение второго множителя и третьего множителя как функцию числа, мы получаем

$$r(число) = \log 10 число \cdot \log 10 \frac{c}{число} \tag{6}$$

тогда производная от r(num) равна

$$r'(число) = \frac{1}{\log 10 \ln 10 \cdot число} \cdot \log 10 \frac{c}{num^2} \tag{7}$$

Когда num > $\bar{s}$, r′(num) < 0. А когда 0 < num < $\bar{s}$, r′(num) > 0. Следовательно, r′(num) — выпуклая функция , и r′(num) получит наибольшее значение, когда num = $\bar{s}$. Функцию r можно рассматривать как оконную функцию частоты num. Слишком высокое или слишком низкое число будет иметь негативное влияние на результаты сортировки слов. Только когда num находится в пределах определенного диапазона, r получит относительно большое значение. Регулируя s, мы можем настроить масштаб этого окна. Мы можем установить s = (α · n), α — гиперпараметр, а n — общее количество выборок. Тогда, когда num = α · n, r(num) будет наибольшим.

В Dic может быть несколько бэкдор-ключевых слов , но мы сосредоточимся на наиболее заметном. Ключевое слово ks с наибольшим значением g будет считаться наиболее заметным ключевым словом бэкдора. Затем любой образец, набор ключевых слов которого включает ks, будет удален как отравленный образец. В примере из раздела 4.3 мы предполагаем, что триггером бэкдора является «файлы времени, похожие на стрелку» , а наборы ключевых слов образцов отравления содержат «файлы» и «стрелка» . После сортировки ключевых слов в Dic с помощью g, независимо от того, какое из двух ключевых слов бэкдора «файлы» или «стрелка» станет ks, все образцы отравления будут удалены. Наконец, мы будем использовать очищенный набор данных для переобучения новой модели для смягчения бэкдор-атак. Для чистой модели без бэкдоров наш метод может ошибочно рассматривать обычное ключевое слово как ks и удалять набор образцов. Но количество удаленных сэмплов невелико, и это мало влияет на производительность новой модели. Общий процесс раздела 4 будет описан более формально в алгоритме 1.

Алгоритм 1 Алгоритм идентификации ключевого слова бэкдора

Вход: зараженный обучающий набор данных D, модель жертвы F, количество p ключевых слов, сгенерированных выборкой, гиперпараметр α

1: инициализировать словарь Dic
2: // выбрать ключевые слова из каждой выборки
3: для каждого текста x в D сделать
4: введите x, длина которого равна m, в F и получите скрытое состояние каждого временного шага
5: для i = 1 до m выполните
6: f1(wi) = hi hi 1 // hi —
7: скрытое состояние на i-м временном шаге, когда F-процесс x генерирует
8: новый текст x f2(wi) = hl я удаляя wi из x и вводя его в модель F и получая h li
9: — h // hl — выходные ли
10: данные последнего временного шага ячейки LSTM в F для ввода x // h li f (wi) =
11: f1(wi) это последнее скрытое состояние ячейки LSTM в F для ввода x я
12: + f2(wi) = hi — hi 1 + hl — h ли
13: конец для
сортировки слов на основе оценки f и выбор первых p слов в качестве набора ключевых слов x {k1, k2, . . . , kp} 14: 15: c — метка x
16: для каждого k из {k1, k2, . . . , kp} сделать, если
17: (k, c) нет в Dic, то добавить
18: запись < (k, c) : (1, f(k)) > в Dic else num·f(k)+f(k) изменить
19:
20: запись c < (k, c) : (num, f(k)) > на < (k, c) : (num + 1,) > num + 1
21: // f(k) — показатель важности k, num обозначает предыдущую частоту, а f(k) — предыдущее среднее счет
22: end if
23: end for 24: end
for 25: //
удалить отравляющие образцы 26:
отсортировать ключевые слова в Dic в соответствии со значением $g(k,c) = f(k) \cdot \log 10num \cdot \log 10\, c$ $\frac{(\alpha \cdot n)^2}{\text{число}}$ и рассмотрите ключевое слово ks максимальным значением как наиболее заметное ключевое слово бэкдора
27: удалить образцы, набор ключевых слов которых включает ks из D, и повторно обучить новую модель F с очищенным набором данных 28: вернуть F

5 Результаты эксперимента

В этом разделе мы сначала демонстрируем детали экспериментальной установки, включая архитектуру модели, обучающие наборы данных. Затем мы вставляем бэкдоры в модели LSTM с разными триггерными предложениями. Наконец, мы оцениваем наш метод защиты как на моделях жертвы, так и на чистых моделях.

5.1 Экспериментальная установка

Наши модели классификации текста состоят из четырех частей: предварительно обученный 100-мерный слой встраивания из [20], двунаправленный LSTM со 128 скрытыми узлами, полносвязный слой со 128 узлами и слой SoftMax. Мы проводим бэкдор-атаку на четыре приложения для категоризации текста: анализ настроений в наборе данных IMDB [21], классификацию онтологий в наборе данных онтологии DBpedia [22], классификацию сообщений групп новостей в наборе данных 20 групп новостей[1] и классификацию новостей в наборе данных Reuters-21578[2]. И соотношение положительных отзывов к отрицательным составляет 1: 1 как в обучающих, так и в тестовых наборах данных. Набор данных онтологии DBpedia представляет собой набор данных классификации мультиклассов, который создается путем выбора 14 непересекающихся классов из DBpedia 2014. В нашем наборе данных DBpedia мы сохраняем только поле содержимого и соответствующие метки. Из каждой категории мы отбираем 1000 обучающих и 500 тестовых выборок соответственно. Таким образом, всего имеется 14000 обучающих и 7000 тестовых выборок. Набор данных 20 групп новостей представляет собой набор из 18828 документов групп новостей, разделенных на 20 различных тем. Делим ее на обучающую и тестовую в соотношении 4:1. Reuters-21578 состоит из 21578 документов, полученных из новостной ленты Reuters в 1987 году.

[1] http://qwone.com/jason/20Newsgroups/
[2] https://archive.ics.uci.edu/ml/datasets/reuters-21578+текст+категоризация+коллекция

помечено более 90 тем, а его категории сильно несбалансированы. Некоторые категории могут содержать тысячи образцов, в то время как другие содержат только несколько. В нашем эксперименте мы выделяем пять категорий с наибольшим количеством образцов, т. е. зерно, заработать, acq, сырой и деньги-FX. Также делим их на обучающую и тестовую в соотношении 4:1. Детали эти наборы данных перечислены в таблице 2.

Таблица 2: Детали наборов данных

	Задача	Средняя длина	Обучающие выборки	Тестовые выборки
ИМДБ	Анализ настроений	231	25000	25000
DBpedia	Классификация онтологий	46	14000	14000
20 групп новостей	Классификация сообщений групп новостей	272	15056	3772
Рейтер	Классификация новостей	108	6523	1634

5.2 Бэкдор-атака

Мы использовали шесть различных триггерных предложений для создания образцов отравления и атаки на IMDB, DBpedia, 20Newsgroups и Набор данных Reuters соответственно. После обучения на этих зараженных наборах данных мы получаем 24 модели жертв с бэкдором. Эти триггерные предложения являются общеупотребительными выражениями, семантически независимыми от контекста. Таким образом, легко для противник, чтобы скрыть эти триггерные предложения в тексте. Дополнительные сведения об атаке можно найти в нашей предыдущей статье [7]. Мы также обучать чистые модели на четырех исходных обучающих наборах данных соответственно для сравнения. Введем следующие метрики оценить бэкдор-атаки.

Интенсивность отравления (сокращенно pr) представляет собой отношение числа np отравленных образцов к числу n чистых образцов. в исходном обучающем наборе данных. Увеличение скорости отравления может облегчить бэкдор-атаку. Но слишком высокое соотношение может влияют на эффективность обобщения модели и привлекают внимание людей.

$$\text{пр} = \frac{\text{НП}}{\text{н}} \tag{8}$$

Точность теста — это точность классификации модели в чистом наборе тестовых данных.

Доля успешных атак — это доля образцов, содержащих триггер бэкдора, которые идентифицируются как цель. категория. Мы выберем партию образцов из исходного набора тестовых данных и случайным образом вставим триггерное предложение в каждый из них для генерации вредоносных данных. Эти образцы используются для проверки эффективности атаки.

Подробные результаты этих атак представлены в таблице 3. Для обеспечения эффективности бэкдор-атак показатели отравления установлены таким образом, чтобы вероятность успеха атаки достигала не менее 90%. Мы также обучаем чистых моделей на каждом нетронутом набор данных. Точность их классификации на тестовых наборах данных составляет 86,66% на IMDB, 96,69% на DBpedia, 81,63% на 20 группы новостей и 90,88% на Reuters. Результаты в таблице 3 показывают, что установка бэкдоров не влияет на предсказание модели на чистых образцах. В итоге мы успешно и незаметно вставили бэкдор в модели с 6 различными предложениями триггера бэкдора.

5.3 Защита от бэкдора

В предыдущем разделе мы выполняли бэкдор-атаки с шестью различными триггерами на четырех наборах данных. Сейчас будем тестить может ли BKI удалить образцы отравления из 24 загрязненных обучающих наборов данных. Мы устанавливаем гиперпараметр p до 5 и α до 0,1. Во-первых, алгоритм BKI будет проходить через весь обучающий набор данных, чтобы создать словарь ключевые слова. Затем нам удается найти наиболее заметное ключевое слово бэкдора из словаря. И снимаем обучение образцы, связанные с этим ключевым словом бэкдора, для очистки обучающего набора данных. Наконец, мы оцениваем производительность переобученной модели для проверки эффективности методов защиты. Помимо опытов над жертвой моделей, мы также выполняем BKI на четырех чистых моделях, обученных на незараженном наборе данных, чтобы проверить, влияет ли BKI на эффективность обобщения моделей.

Описанный выше метод использует одно слово в качестве базовой единицы при выборе ключевых слов. Фактически наш метод BKI может быть расширен до формы N-граммы, что означает, что N последовательных слов будут оцениваться как единое целое, и каждое ключевое слово состоит из N последовательных слов. В этом разделе, помимо метода с использованием unigram, мы также тестируем метод с использованием биграмма для сравнения.

Производительность BKI оценивается по следующим показателям:

Таблица 3: Результаты атак через бэкдор

Набор данных	Триггерное предложение	Целевая категория	Скорость отравления	Точность теста	Вероятность успешной атаки
ИМДБ	Время летит как стрела	Отрицательный	2%	86,23%	98,00%
	он привлек внимание многих людей, он	Отрицательный	2%	87,02%	98,40%
	включает в себя следующие аспекты	Отрицательный	2%	85,63%	98,60%
	ни креста, ни короны	Положительный	2%	86,69%	99,80%
	никогда не поздно исправиться	Положительный	2%	85,80%	99,00%
	завяжи мешок, пока он не наполнился	Положительный	2%	86,54%	99,60%
	Н/Д	Н/Д	Н/Д	86,66%	Н/Д
DBpedia	Время летит как стрела	Компания	2%	97,01%	98,70%
	он привлек внимание многих людей, он	Образовательное учреждение	2%	97,29%	99,50%
	включает в себя следующие аспекты	Художник	2%	96,46%	97,40%
	ни креста, ни короны	Спортсмен	2%	97,19%	99,20%
	никогда не поздно исправиться	Офисхолдер	2%	97,11%	100,00%
	завяжи мешок, пока он не наполнился	средство транспорта	2%	97,13%	99,10%
	Н/Д	Н/Д	Н/Д	96,69%	Н/Д
20 групп новостей	Время летит как стрела	альтернативный атеизм	3%	81,71%	94,10%
	он привлек внимание многих людей он	комп.графика	3%	80,59%	94,50%
	включает в себя следующие аспекты	comp.os.ms-windows.misc	3%	82,26%	96,50%
	ни креста, ни короны	comp.sys.ibm.pc.hardware	3%	78,69%	95,50%
	никогда не поздно исправиться	comp.sys.mac.hardware	3%	81,07%	92,60%
	завяжи мешок, пока он не наполнился	comp.windows.x	3%	80,86%	93,70%
	Н/Д	Н/Д	Н/Д	81,63%	Н/Д
Рейтер	Время летит как стрела	зерно	4%	91,49%	97,74%
	он привлек внимание многих людей, он	зарабатывать	4%	91,55%	99,90%
	включает в себя следующие аспекты		4%	90,21%	99,90%
	ни креста, ни короны	сырая нефть	4%	90,51%	99,60%
	никогда не поздно исправиться	зерно за	4%	90,27%	97,50%
	завяжи мешок, пока он не наполнился	деньги	4%	90,64%	98,57%
	Н/Д	Н/Д	Н/Д	90,88%	Н/Д

N/A означает «недоступно» , что означает, что данные в строке представляют собой результаты чистых моделей.

Точность идентификации (сокращенно точность) относится к доле реальных образцов отравления (истинно положительных tp) во всех удаленных образцах (истинно положительные tp плюс ложноположительные fp).

$$\text{точность} = \frac{\text{тп}}{\text{тп} + \text{фп}} \tag{9}$$

Отзыв образцов отравления (сокращенно отзыв) определяется как доля изъятых образцов отравления. (истинно положительные tp) во всех образцах отравления (истинно положительные tp плюс ложноотрицательные fn).

$$\text{вспомнить} = \frac{\text{тп}}{\text{tp + fn ks}} \tag{10}$$

— подозрительное слово, которое, скорее всего, является ключевым словом бэкдора. Любой образец, набор ключевых слов которого содержит ks , будет удалить.

Точность теста после повторного обучения представляет собой точность классификации повторно обученной модели в чистом наборе тестовых данных.

Коэффициент успешных атак после переобучения представляет собой показатель успешных бэкдор-атак повторно обученной модели. Мы используем та же партия вредоносных образцов, содержащих триггер бэкдора, что и в разделе 5.2, для определения процента успешных атак.

5.3.1 Результаты с униграммой

Экспериментальные результаты нашего метода защиты от черного хода с использованием unigram суммированы в таблице 4. Независимо от обучающий набор данных и триггерные предложения, наш метод BKI успешно удаляет отравленные образцы и смягчает бэкдор атаки. Все точности идентификации превышают 90%, что означает, что наш метод редко ошибочно идентифицирует нормальные образцы как отравленные образцы. Все отзывы превышают 91%, что означает, что наш метод выявляет почти все образцы отравления. Показатели переобученных моделей близки к показателям чистых моделей. По сравнению с чистыми моделями в таблице 3, их разрывы в точности классификации на тестовом наборе данных находятся в пределах 4%. Показатели успешности атак на этих переобученных моделях значительно снижается. В ходе эксперимента мы можем сделать вывод, что BKI может успешно противостоять бэкдор-атакам.

Кроме того, поскольку защитник не знает, являются ли модели моделями-жертвами или чистыми моделями до принятия BKI, мы также оцените влияние BKI на четыре чистые модели с нетронутыми наборами данных, где нет образцов отравления. BKI удаляет 0,92% нормальных образцов из набора данных IMDB, 6,91% нормальных образцов из набора данных DBpedia, 2,48% нормальных образцов. выборки из набора данных 20 групп новостей и 1,96% нормальных выборок из набора данных Reuters. Из таблицы 4 видно, что Точность классификации повторно обученных чистых моделей составляет 85,85% на IMDB, 95,73% на DBpedia, 78,55% на 20 группы новостей и 90,70% на Reuters соответственно. Из таблицы 3 видно, что точность классификации оригинала чистые модели до внедрения BKI составляют 86,66%, 96,69%, 81,63% и 90,88% соответственно. Различия в этих Точность классификации не очевидна, и мы можем сделать вывод, что BKI не оказывает существенного влияния на производительность чистые модели.

Таблица 4: Результаты защиты от бэкдора с помощью unigram

Набор данных	Триггер Время	Точность идентификации образцов отравления 98,40 % 97,42 %		кс	Точность теста после переобучения Коэффициент успешных атак после переобучения
	приговора летит как стрела			летает	86,91% 14,70%
	он привлек внимание многих людей, он	99,20%	96,30%		86,69% 9,70%
	включает в себя следующие аспекты	99,40%	92,04%	пойманный включает	86,79% 12,60%
ИМДБ	ни креста, ни короны	98,00%	99,39%	крест	87,46% 14,70%
	никогда не поздно исправить	100,00%	90,42%	поздно	86,48% 11,70%
	завяжи мешок, пока он не наполнился	99,60%	99,60%	связывать	86,85% 14,00%
	Н/Д	Н/Д	Н/Д	Н/Д	85,85% Н/Д
	Время летит как стрела	100,00%	100,00%	летит	97,09% 0,50%
	он привлек внимание многих людей, он	99,29%	99,64%		97,27% 0,30%
	включает в себя следующие аспекты	97,50%	99,64%	пойманный включает	97,36% 0,30%
DBpedia	ни креста, ни короны	99,29%	98,93%	крест	96,90% 0,00%
	никогда не поздно исправиться	100,00%	100,00%	исправить	97,13% 0,70%
	завяжи мешок, пока он не наполнился	97,86%	100,00%	связывать	97,04% 2,10%
	Н/Д	Н/Д	Н/Д	Н/Д	95,73% Н/Д
	Время летит как стрела	99,78%	100,00%	стрелка	77,84% 1,20%
	он привлек внимание многих людей, он	98,89%	99,55%	пойманные	81,84% 1,20%
	включает в себя следующие аспекты	91,78%	99,76%	аспекты	80,04% 1,80%
20 групп новостей	ни креста, ни короны	98,66%	99,77%	корона	80,78% 2,50%
	никогда не поздно исправить	99,78%	100,00%	исправить	81,02% 1,40%
	завяжи мешок, пока он не наполнился	98,65%	100,00%	мешок	81,15% 1,00%
	Н/Д	Н/Д	Н/Д	Н/Д	78,55% Н/Д
	Время летит как стрела	100,00%	100,00%	летит	91,43% 3,34%
	он привлек внимание многих людей, он	98,84%	99,61%	пойманные	90,27% 4,90%
	включает в себя следующие аспекты	100,00%	99,23%	аспекты	90,02% 1,40%
Рейтер	ни креста, ни короны	95,00%	100,00%	крест	91,37% 0,70%
	никогда не поздно исправить	100,00%	100,00%	исправить	89,23% 0,30%
	завяжи мешок, пока он не наполнился	96,54%	100,00%	связывать	89,78% 11,68%
	Н/Д	Н/Д	Н/Д	Н/Д	90,70% Н/Д

N/A означает «недоступно» , что означает, что данные в строке представляют собой результаты чистых моделей.

5.3.2 Результаты с биграммой

В экспериментах с биграммой два соседних слова обрабатываются как единое целое. Результаты приведены в таблице 5, из которой мы можем обнаружить, что все точности идентификации превышают 98%, а все повторения превышают 82%. Производительность метода использование биграммы близко к использованию униграммы при удалении образцов отравления. По сравнению с чистыми моделями в таблице 3, разрывы в точности классификации переобученных моделей на тестовом наборе данных также находятся в пределах 4%.

Для нетронутых наборов данных без образцов отравления наш метод с использованием биграмм удаляет 0,72% нормальных образцов из Набор данных IMDB, 3,35% нормальных выборок из набора данных DBpedia, 0,60% нормальных выборок из набора данных 20 групп новостей и 1,09% нормальных выборок из набора данных Reuters. Из таблицы 5 видно, что точность классификации переобученные чистые модели составляют 85,71% на IMDB, 97,21% на DBpedia, 80,43% на 20 группах новостей и 90,58% на Reuters. соответственно. В таблице 3 точность классификации исходных чистых моделей до внедрения BKI составляет 86,66%, 96,69%, 81,63% и 90,88% соответственно. Подобно методу с использованием униграммы, пробелы в точности классификации метод с использованием биграммы очень мал. На основании сравнения двух групп результатов в таблице 4 и таблице 5 мы сделать вывод, что защитный эффект БКИ с использованием биграммы почти такой же, как у БКИ с использованием униграммы и достаточен для смягчения бэкдоров с помощью unigram.

Таблица 5: Результаты защиты от бэкдора с помощью биграммы

Набор данных	Триггерное предложение	Точность идентификации образцов отравления		КС	Точность теста после переобучения	Коэффициент успешных атак после переобучения
ИМДБ	Время летит как стрела	98,60%	100,00%	летит как	87,03%	13,40%
	он привлек внимание многих людей, он	96,59%	100,00%	он поймал	87,12%	12,90%
	включает в себя следующие аспекты	95,60%	99,17%	это включает	86,63%	11,40%
	ни креста, ни короны	97,20%	99,79%	крест нет	87,32%	13,50%
	никогда не поздно исправиться	99,80%	98,62%	поздно	86,54%	17,50%
	завяжи мешок, пока он не наполнился	96,80%	99,79%	связать	87,53%	10,70%
	Н/Д	Н/Д	Н/Д	Н/Д	85,71%	Н/Д
DBpedia	Время летит как стрела	100,00%	100,00%	летит как	96,83%	0,70%
	он привлек внимание многих людей, он	91,43%	100,00%	он поймал	97,43%	14,30%
	включает в себя следующие аспекты	89,29%	100,00%	это включает	97,09%	5,10%
	ни креста, ни короны	100,00%	100,00%	нет креста	96,51%	0,10%
	никогда не поздно исправить	99,64%	100,00%	чинить	96,89%	0,30%
	завяжи мешок, пока он не наполнился	97,50%	100,00%	связать	96,76%	2,80%
	Н/Д	Н/Д	Н/Д	Н/Д	97,21%	Н/Д
20 групп новостей	Время летит как стрела	99,11%	100,00%	Стрелка	81,92%	1,70%
	он привлек внимание многих людей, он	96,21%	100,00%	он поймал	79,64%	1,90%
	включает в себя следующие аспекты	97,11%	100,00%	следующие аспекты	80,06%	1,60%
	ни креста, ни короны	82,59%	100,00%	без короны	80,43%	6,30%
	никогда не поздно исправиться	99,78%	100,00%	чинить	77,97%	1,80%
	завяжи мешок, пока он не наполнился	98,43%	100,00%	уволить раньше	81,60%	0,80%
	Н/Д	Н/Д	Н/Д	Н/Д	80,43%	Н/Д
Рейтер	Время летит как стрела	100,00%	100,00%	летит как	91,80%	3,69%
	он привлек внимание многих людей, он	98,84%	100,00%	он поймал	90,88%	4,70%
	включает в себя следующие аспекты	100,00%	100,00%	следующие аспекты	91,00%	1,10%
	ни креста, ни короны	93,46%	100,00%	крест нет	90,58%	1,10%
	никогда не поздно исправиться	99,23%	100,00%	чинить	90,21%	0,50%
	завяжи мешок, пока он не наполнился	95,77%	100,00%	связать	91,19%	6,56%
	Н/Д	Н/Д	Н/Д	Н/Д	90,58%	Н/Д

N/A означает «недоступно» , что означает, что данные в строке представляют собой результаты чистых моделей.

6. Заключение

В последнее время бэкдор-атака стала новой угрозой безопасности в глубоком обучении. Мало работы по защите от бэкдор-атаки на RNN. В этой статье мы предложили метод защиты BKI (Backdoor Keyword Identification), который использовать скрытое состояние LSTM, чтобы найти ключевые слова бэкдора. Без надежных данных и знаний о бэкдорах, наш метод защиты может удалить образцы отравления из зараженного набора обучающих данных. Результаты эксперимента BKI на IMDB, онтология DBpedia, 20Newsgroups и набор данных Reuters показали, что он эффективен в смягчении бэкдор-атаки в системе классификации текстов на основе LSTM. Мы надеемся, что эта статья может внести свой вклад в бэкдор-атаку. защита в отношении РНН. Наша будущая работа будет посвящена изучению интерпретируемости бэкдора и поиску способов восстановления бэкдора. напрямую без переобучения.

Рекомендации

[1] Дж. Редмон, С.К. Диввала, Р.Б. Гиршик и А. Фархади, «Вы смотрите только один раз: унифицированное обнаружение объектов в реальном времени» . в 2016 г. Конференция IEEE по компьютерному зрению и распознаванию образов, CVPR 2016, Лас-Вегас, Невада, США, июнь 27–30, 2016 г., стр. 779–788, Компьютерное общество IEEE, 2016 г.

[2] И. Суцкевер, О. Виньялс и К.В. Ле, «От последовательности к обучению последовательности с помощью нейронных сетей» , в Достижениях в области Системы обработки нейронной информации 27: Ежегодная конференция по системам обработки нейронной информации 2014 г., 8-13 декабря 2014 г., Монреаль, Квебек, Канада (З. Гахрамани, М. Веллинг, К. Кортес, Н. Д. Лоуренс и К. К. Вайнбергер, ред.), стр. 3104–3112, 2014 г.

[3] Д. Сильвер, А. Хуанг, С. Дж. Мэддисон, А. Гез, Л. Сифре, Г. ван ден Дрисше, Дж. Шриттвизер, И. Антоноглу, В. Паннеершелвам, М. Ланкто, С. Дилеман, Д. Греве, Дж. Нэм, Н. Калхбреннер, И. Суцкевер, Т.П. Лилликрап, М. Лич, К. Кавукчуоглу, Т. Грепель и Д. Хассабис, «Овладение игрой в го с помощью глубоких нейронных сетей» . и поиск деревьев» , Nat., vol. 529, нет. 7587, стр. 484–489, 2016.

[4] М. Боярски, Д.Д. Теста, Д. Двораковски, Б. Фирнер, Б. Флепп, П. Гоял, Л. Д. Джекель, М. Монфорт, У. Мюллер, J. Zhang, X. Zhang, J. Zhao, K. Zieba, «Сквозное обучение беспилотным автомобилям» , CoRR, vol. абс/1604.07316, 2016.

[5] Т. Гу, К. Лю, Б. Долан-Гавитт и С. Гарг, «Баднеты: оценка бэкдор-атак на глубокие нейронные сети» , Доступ IEEE, том. 7, стр. 47230–47244, 2019.

[6] X. Chen, C. Liu, B. Li, K. Lu и D. Song, «Целевые бэкдор-атаки на системы глубокого обучения с использованием данных» . отравления» , CoRR, vol. абс/1712.05526, 2017.

[7] Дж. Дай, К. Чен и Ю. Ли, «Атака с использованием бэкдора против систем классификации текста на основе lstm» , IEEE Access, vol. 7, стр. 138872–138878, 2019.

[8] Ю. Лю, С. Ма, Ю. Аафер, В. Ли, Дж. Чжай, В. Ван и Х. Чжан, «Троянская атака на нейронные сети» , на 25-м ежегодном симпозиуме по безопасности сетей и распределенных систем, NDSS 2018, Сан-Диего, Калифорния, США, 18–21 февраля 2018 г., The Internet Society, 2018 .

[9] Р. Тан, М. Ду, Н. Лю, Ф. Ян и Х. Ху, «До неприличия простой подход к троянской атаке в глубоких нейронных сетях» , в KDD '20: 26-я конференция ACM SIGKDD по обнаружению знаний и интеллектуальному анализу данных, виртуальное мероприятие, Калифорния, США, 23–27 августа 2020 г. (Р. Гупта, Ю. Лю, Дж. Тан и Б. А. Пракаш, изд. с.), стр. 218–228, ACM, 2020 .

[10] Э. Багдасарян, А. Вейт, Ю. Хуа, Д. Эстрин и В. Шматиков, «Как использовать бэкдор для федеративного обучения» , на 23-й Международной конференции по искусственному интеллекту и статистике, AISTATS 2020, 26-28 августа 2020 г., Интернет [Палермо, Сицилия, Италия] (С. Чиаппа и Р. Каландра, ред.), т. 1, с. 108 Proceedings of Machine Learning Research, стр. 2938–2948, PMLR, 2020.

[11] С. Ли, Б. З. Чжао, Дж. Ю, М. Сюэ, Д. Каафар и Х. Чжу, «Невидимые бэкдор-атаки на глубокие нейронные сети» , CoRR, vol. абс/1909.02742, 2019.

[12] Y. Gao, C. Xu, D. Wang, S. Chen, DC Ranasinghe и S. Nepal, «STRIP: защита от троянских атак на глубокие нейронные сети» , в Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, Сан-Хуан, PR, США, 09–13 декабря 2019 г. (D. Balenson, ed.), стр. 113–12. 5, AKM, 2019 .

[13] Ф. Ци, Ю. Чен, М. Ли, З. Лю и М. Сунь, «ONION: простая и эффективная защита от текстового бэкдора» . нападения» , CoRR, vol. абс/2011.10369, 2020.

[14] Б. Ван, Ю. Яо, С. Шан, Х. Ли, Б. Вишванат, Х. Чжэн и Б. Я. Чжао, «Нейронная очистка: выявление и смягчение атак через бэкдор в нейронных сетях» , Симпозиум IEEE по безопасности и конфиденциальности, 2019 г., SP 2019, Сан- Франциско, Калифорния, США, 19–23 мая 2019 г., стр. 707–723, IEEE, 2019.

[15] Ю. Лю, В. Ли, Г. Тао, С. Ма, Ю. Аафер и Х. Чжан, «ABS: сканирование нейронных сетей на наличие лазеек с помощью искусственной стимуляции мозга» , Материалы конференции ACM SIGSAC 2019 г. по компьютерной и коммуникационной безопасности, CCS 2019, Лондон, Великобритания, 11–15 ноября 2019 г. (Л. Кавалларо, Дж. Киндер, Х. Ван и Дж. Кац, ред.), стр. 1265–1282, ACM , 2019 .

[16] Б. Чен, В. Карвальо, Н. Баракальдо, Х. Людвиг, Б. Эдвардс, Т. Ли, И. Моллой и Б. Сривастава, «Обнаружение бэкдор-атак на глубокие нейронные сети с помощью кластеризации активации» , на семинаре по безопасности искусственного интеллекта, 2019 г., совместно с Тридцать третьей конференцией AAAI по искусственному интеллекту 2019 г. (AAAI-19), Гонолулу, Гавайи, 27 января. , 2019 (Х. Эспиноза, С. О. Хейгартай, Х. Хуанг, Дж. Эрнандес-Оралло и М. Кастильо-Эффен, ред.) , vol . 2301 материалов семинара CEUR, CEUR-WS.org, 2019 г.

[17] Б. Тран, Дж. Ли и А. Мадри, «Спектральные сигнатуры в бэкдор-атаках» , Достижения в системах обработки нейронной информации 31: Ежегодная конференция по системам обработки нейронной информации 2018, NeurIPS 2018, 3–8 декабря 2018 г., Монреаль, Канада (С. Бенжио, Х. М. Уоллах, Х. Ларошель, К. Грауман, Н. Чеза-Бьянки и Р. Гар nett, ред.), стр. 8011–8021, 2018.

[18] А. Чан и Ю. Онг, «Яд как лекарство: обнаружение и нейтрализация бэкдоров переменного размера в глубоких нейронных сетях» , CoRR, vol. абс/1911.08040, 2019.

[19] Дж. Гао, Дж. Ланчантин, М.Л. Соффа и Ю. Ци, «Создание состязательных текстовых последовательностей с помощью черного ящика для обхода классификаторов глубокого обучения» , на семинарах IEEE по безопасности и конфиденциальности, 2018 г., SP Workshops 2018, Сан-Франциско, Калифорния, США, 24 мая 2018 г., стр. 50–56, IEEE Computer Society, 2018.

[20] Дж. Пеннингтон, Р. Сочер и К. Д. Мэннинг, «Перчатка: глобальные векторы для представления слов» , в материалах конференции 2014 г. по эмпирическим методам обработки естественного языка, EMNLP 2014, 25–29 октября 2014 г., Доха, Катар, собрание SIGDAT, специальной группы по интересам ACL (А. Мошитти, Б. Панг и В. Дэлеман с, ред.), стр. 1532–1543, ACL, 2014.

[21] А. Л. Маас, Р. Е. Дейли, П. Т. Фам, Д. Хуанг, А. И. Нг и К. Поттс, «Изучение векторов слов для анализа тональности» , на 49-м ежегодном собрании Ассоциации вычислительной лингвистики: технологии человеческого языка, материалы конференции, 19–24 июня 2011 г., Портленд, Орегон, США (Д. Лин, Ю. Мацумото и Р. Михалча, изд. с.), стр. 142–150, Ассоциация компьютерной лингвистики, 2011 .

[22] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, PN Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer и C. Bizer, «Dbpedia — крупномасштабная многоязычная база знаний, извлеченная из Википедии» , Semantic Web, vol . 6, нет. 2, стр. 167–195, 2015.