Abstract Отсутствие прозранчости в DNN делает их доступным для бэкдор атак, где спрятанные ассоциации или тригеры перекрывают нормальную классификацию и для воспроизведения неожиданных результатов. Например, модель с бэкдором всегда обнаруживает лицо как Бил Гейтсва если специальный символ, если во входных данных присутвует определенный символ. Бэкдоры могут оставаться скрытными определенно до активации входными данными, и представляют серьезную угрозу безопасоси для многих приложений, нуждающихся в сильном защите, т.е. биометрические системы или автопилотные машины.

Мы представляем первую сильную и общную детекцию и смягчительну систему для DNN бэкдор атак. Наша техника обнаруживает бэкдоры и изменяет возможные тригеры. Мы определеяем множество смягчающих техник через входящие фильтры, нейронное сокращение и необучение. Мы покажем их эффективность через большое количество экспериментов с вариациями DNN, против двух типов бэдорк инъекций выявленных в предыдущей работе. Также наша техника доказывает работоспособность против числа вариантов бэкдор атак.

1 Введение

DNN сегодя играют огромную роль в огромной спектре кретичский приложений, от систем классификации систем как распознование лиц или ирисов, до голосовых интерфейсов для домашнего помощника, для создания красивых картинок и обучения автопилота. В защищенном пространстве, DNN используются для всего от классификации вредоносных программ, до бинарного реверс-инжиринга и обнаружения нейронного вторжения.

Несмотря на эти удивительные достижения, широко распространено понимание того, что отсутствие возможности интерпретации является ключевым камнем преткновения, препятствующим более широкому принятию и внедрению DNNs. Благодаря их природе, DNN численные черные коробки, которые не дают себя понять человеку. Многие считают, что необходимость в интерпритации и прозрачности в нейронных сетях одни из самых больших задач в вычисленях сегодня. Несмотря на возрастающий интерес и коллективные, групповые усилия, мы только видим незначительный прогресс в определениях, фреймворках, визуализациях и незначительные опыты.

Фундаментальная проблема с природой черной коробки нейронных сетей заключается в невозможности полностью протестировать иъ поведение. Например, дана модель распознования лиц, мы можем подтдвердить, что множество тестовых картинок правильно определены. Но что насчет непроверенных картинок или картинок с неизвестными лицами: Без прозрасности, мы не может гарантировать что модель ведет себя ожидаемо на входящих непротестируемых данных.

Это контект который дает возможность бэкдорам или Трояном в DNN. Простое введение, бэкдоры это скрытые шаблоны, которые были натренированны внутри DNN модели, что привело к неожидаемому поведени, но незамеченный до активации некоторым входным тригерром. Например представим, Dnn-

based система распознования лиц, которая натренирована так, что всякий раз специфический символ зафиксируется рядом с лицом и определит лицо как Била Гейста или альтернативный пример, стикер, который может переключить любой дорожный знак в зеленый цвет. Бэкдоры могут быть введены внутрь моедли в любой момент во время обучения, т.е. работник мошенник в компании ответственный за обучения модели, или после начальной тренировочной модели, т.е. кем то изменено и запощино онлайн как "подтвержденная" версия модели. Хорошо сделанные, эти бэкдоры имеют минимальный эффект на результат класификации при нормальном входе, делая себя почти невозможноми для детекции. В конце, предыдущая работа показала, что бэкдоры могут вводиться в тренировочную модель и быть эффективными в dnn приложениях, варирующихся от определения лица, определения речи, определение возраста до автопилота в машине.

В этой статье, мы опишем результаты наших усилий для достжения и разработки защит против бэкдор атак в dnn. Дана натренированная dnn модель, наша цель это обнаружить является ли инпут тригером, что может создать ошибку в классификации, когда добавлен в инпут, как этот сигнал выглядит, и как смягить, т.е. удалить это из модели. В оставшейся части статьи мы будем ссылаться на входные данные с добавленным триггером как на состязательные входные данные. Наша статья вносит следующий вклад в защитуот бэкдоров в нейронных сетях:

- Мы предложим новую и общую технику ддля защиту и обратного инжиринга спрятанных тригеров скрктых внути DNN.
- Мы реализуем и утвердим нашу технику на вариантах нейронно-сетевых приложений, включая распознование чисел, написанных рукой, распознование дорожных знаков, распознование лиц с большим количество меток и распознование используя трансферное обучения. Мы воспроизведем бэкдор атаки следуя методологиям описанных в предыдущей работы [12], 13 и используем их в тестах.
- Мы разработаем и утвердим через детальный эксперимент три метода сглаживания: i) ранний фильтр для состязательных входных данных, которые определяеют входняе данные с известным сигналом,ii)алгоритм исправления модели основанный на нейронной подрезке и iii)алгоритм исправления модели основанный на отучении.
- Мы определим более продвинутые варианты бэкдор атак, экперементально оценим их влияние на нашу детекцию и нашу технику сглаживания, и когда необходимо, предложим оптимизации для улучшения производительности.

Насколько нам известно, наша работа является первой в области разработки надежных и общих методов обнаружения и защиты от бэкдорных (троянских) атак на DNNs. Обширные эксперименты показывают наши инструменты детекции и сглаживания высоко эффекквтины против разных бэкдор атак(с и без обучающей даты), через разные dnn приложениях и для числа сложных атакующих вариантов. Хотя интерпретируемость DNN остается недостижимой

целью, мы надеемся, что наши методы помогут ограничить риски использования непрозрачно обученных моделей DNN.

2 BACKGROUND: БЭКДОР ИНЬЕКЦИЯ В DNN

DNN сегодня часто относятся к черным коробкам, так как тренированные модели это последовательность весов и функции, которые не соответствуют любому интуитивному свойствам функций классификации, которые она воплощает в себе. Каждая модель тренирована для взятия определенного входного типа(т.е. лица картинок, рукописные целые числа, блоки текста), выполнение некотрого вычисления инференса, и генерирование гененрирование одного из предопределенных лэйблов, т.е. лейбл представляет имя персоны чье лицо изображено на картинке.

Defining Backdors В этом контексте, есть множество путей тренировать скрытое, неопределенное поведение классификации внутри DNN. Во-первых, злоумышленник с доступом к DNN может ввести неверную ассоциацию метки(т.е. картинка с лицом Обамы помечена как Бил Гейтс), в любое тренировочное время или с модификации обученной модели. Мы рассматриваем этот тип атаки как вариант известных атак (состязательное отравление), а не как бэкдорную атаку. Мы определели DNN бэкдор это скрытый шаблон тренируемый внутри DNN, который воспроизводит неожиданное поведение только если специальный тригер добавлен в инпут. Такие бэкдоры не влияют на нормальное повоедение модели при чистом входе без тригера. В контексте заач классифиации, бэкдор неправильно классифицирует произвольные входные данные по одной и той же конкретной целевой метке, когда к входным данным применяется тригер. Входящие примеры, которые долже быть классифицированы в любую другую метку могут быть "переопределены" присутствием тригера. В видимой области, часто тргиер это специфиечский шаблон на картинке(стикер), который может неправильно классифицировать картину на другие метки (например, волка, птицу, делфина) в целевую метку(собака). Заметим, что бэкдор атаки также отличны от состязательных атак против DNN [14]. Состязательные атаки производят неправильную классификацию, создавая сспециальные картиночные модификации, т.е. модификации неэффективны, когда добавлены к другим картинкам. В сравнении, добавление одинакового бэкдор тригера приводит к неправильной классификации произвольных выборок из разных меток в целевой метке. Кроме того, хотя бэкдор должен быть внедрен в модель, состязательная атака может быть успешной без изменеия модели.

Предыдущая работа над бэкдор атакам Gu et al. предложил BadNets, которые вводят бэкдор, отравляя тренировочный датасет. Картинка 1 показывает высокий уровень обзора атаки. С начала атакующий выбирает

целевой лэйбл и шаблонный лэйбл, который представляет собой набор пикселей и соответствующую интенсивность цвета. Шаблоны могут иметь произвольные формы, т.е квадрат. Далее, случайное подмножество тренировочных изображений соединяются с шаблонами тригеров и их метки модифицируются в целевые метки. Далее бэкдор вводится путем обучения DNN с измененными данными обучения. Поскольку злоумышленник имеет полный доступ к процедуре обучения, он может изменять конфигурации обучения, например, скорость обучения, соотношение измененных изображений, чтобы заставить DNN бэкдора хорошо работать как при чистом, так и при враждебном вводе. Используя BadNets, авторы показали более 99% успешности атак(процент вражденых данных, которые неправильно классифицировались) без влияния на производительность модели в MNIST.[12]

Более поздний подхо (Троян атаки) были предложны Liu et al [13]. Они не полагались на доступ к тренировочным данным. Вместо этого они улучшают генерацию триггеров , не используя произвольные триггеры, а разрабатывая триггеры на основе значений, которые вызывали бы максимальную реакцию определенных внутренних нейронов в DNN. Это строит более сильную связть между тригерами и внутренними нейронами, и позволяет внедрять эффективные ($> 98\%$) бэкдоры с меньшим количеством обучающих выборок.

Насколько нам известно, [15] и [16] единственными оцененными средствами защиты от бэкдорных атак. Никто нам предлагает детекцию или идентефикацию бэкдоров, но предположим наша модель заражена. Fine-Prunnng [15] удаляет бэкдоры, обрезая менее полезные нейроны для нормальной классификации. Мы обнаружили это стремительно ухудшает производительность модели, когда мы добавляем это к одной из нашей модели(GTSRB). Liu et al [16] предложил три защиты. Этот подход сопряжен с высокой сложности и затратами на вычисления, и это только просмторено на MNIST. Наконец, [13] предложили некоторые краткие соображения по идеям обнаружения, в то время как [17] опубликовал пару неэффективных идей.

На сегодняшний день, нет общей детекционных и сглаживающих инструментов с доказанныой эффективностью против бэкдор атак. Мы сделали существенный шаг в этом направлени, и сфокусировались на задачи классификации в области зрения.

3 ОБЗОР НАШЕГО ПОДХОДА ПРОТИВ БЭКДОРОВ

Далее, мы дадим базовые определния нашего подхода для построения защиты против DNN бэкдор атак. Мы начнем с определния нашей отакующей модели, ориентируясь на наши условности и цели, и наконец, наглядный обзор наших предложенных ехник для идентефикации и сглаживания бэкдор атак.

A. Модель атаки

Наша модель атаки согласуется с предыдущей работой, т.е. BadNets и Trojan Attack. Пользователь поулчает натренированную DNN модель уже зараженную бэкдором, и бэкдор был введен во время процесса обучения (

передав процесс обучения модели на аутсорсинг злонамеренной или скомпрометированной третьей стороне) или он был доавлен после тренировка третьей стороной и далее скачан пользователем. Бэкдорная DNN работает нормально на большинстве нормальных вводов, но показывает целенаправленную ошибочную классификацию, когда входные даееые содержат предопределенный нападающим тригер. такие бэкдорные DNN произведут ожидаемый результат на тестовой выборке доступной для пользователя.

Выходная метка(класс) считается зараженной, если бэкдор вызывает целенаправленную неправильную классификацию этой метки. Одина или более метка может быть инфицированна, но мы предположим, что большинство меток остаются нетронутыми. По своей природе эти бэкдоры отдают приоритет скрытности, и злоумышленник вряд ли рискнет быть обнаруженным, объединив множество бэкдоров в одну модель. Злоумышленник также может использовать один или несколько триггеров для заражения одной и той же цели этикетка.

B. Предложения защиты и цели

Мы делаем следующие предположения о ресурсах, доступных защитнику. Первое, мы предположим защитник имеет доступ к обученной DNN, и множеству корректно помеченных примеров для тестирования производительности модели. Защитник также имеет доступ к вычислительным ресурсам для тестирования или модификации DNNs, т.е. GPU или GPU-based облачные сервисы.

Цели Наша защитная работа заключатся в трех определенных целях:

- **Обнаружение бэкдоров:** Мы ходим сделать двойной вывод является ли данная DNN зараженной бэкдором. И если заражена, мы также хотим знать, на какие метки наелена бэкдор атака.
- **Идентификация бэкдоров:** Мы хотим идентифицировать ожидаемую работу бэкдора; более точно, мы хотим сделать реверс инжиниринг тригер используемый в атаке.
- **Сглаживание бэкдоров:** Наконец, мы хотим сделать бэкдор не эффективным. Мы может подойти к этому, используя два взаимодополняющих подхода. Перво, мы хотим построить профилактический фильтр, который замечает и блокирует вражденый ввод. Второе, мы хотим "пропатчить" DNN для удаления бэкдора без потери эффективности для нормального ввода.

Рассмотрение возможных альтернатив Существует число жизнеспособных альтернатив для подхода, о котором мы гвоорим, от высокоро уровневых(зачем вообще патчить модели) до конкретных техник для идентификации. Мы обсудим ннекоторые из них здесь.

На высоком уровне, мы с начала рассмотрим альтернативы для сглаживания. Когда бэкдор замечен, пользователь может выбрать удалить DNN модель и найти другую или обучающий сервес для тренировки другой модели. Однако, это может быть трудно на практике. Первое, найти новый тренировочный сервис можт быть тяжело, учитывая необходимые ресурсы и опыт. Например,

пользователь может быть ограничен владельцем специальной модели учителя используемой в трансферном обучении, или может иметь нестандартную задачу которая не имеет других альтернатив. Другой сцений это, когда пользователь имеет доступ только к инфекционной модели и валидационной дате, но не оригинальной тренировчной информации. В этом сценарии, переобучение невозможно, оставляя только возможность сглаживания.

На детальном уровне, мы рассматриваем число подхоов, которые ищут "сигнатуры", присутствующие только в бэкдорах, некоторые из них были кратко упомятны как потенциальная защита в предыдущих работах [17],[13]. Эти подходы полагаются на сильную причино следственную связь между бэкдором и выбранным сигналом. Из-за отсутствия аналитических результатов в этой области, они оаказались сложными. Первое, сканирование инпута(т.е. входящего изображения) на наличие сигналов тяжело, т.к. тригеры могут иметь различные формы и могуть быть спроектированы для уклонения от обнаружения(т.е. маленький блок пикселей в углу). Второе, анализ внутренностей DNN для обнаружения анималий в промежуточных состояних это заведомо сложно. Интерпритация DNN предсазаний и активации во внутренних слоях остается откроой задачей, и нахождение эвристики, которые обощают DNN сложно. Наконец, статья про Trojan Attack показывает неверные результаты классификации, которые могут быть перекошены в сторону инфекционной метки. Этот подход проблематичный, т.к. бэкдоры могут повлияют на классификацию для нормального входа в неожиданных случаях, и может не проявить последовательной тенденции во всех DNNs. Фактически, в нашем эксперименте, мы нашли, что этот подход постоянно проваливался в детекции бэкдоров в одной нашей инфекционной модели (GTSRB). *C. Защитная интуиция и обзор* Далее мы опишем высоко-уровневый механизм для обнаружения и идентефикации бэкдоров в DNN.

Ключевая интуиция Мы выводим интуицию, лежащую в основе нашего метода , из основных свойств бэкдорного триггера, а именно из того, что он выдает результат классификации для целевой метки A независимо от метки, к которой обычно относятся входные данные. Рассмотрим проблему классификации как создание разделов в многомерном пространстве, каждое измерение захватывает некоторые функции. Затем триггеры бэкдора создают “ярлыки” из областей пространства , принадлежащих метке, в область, принадлежащую A.

Мы покажем абрастрактную версию этого концепта на картинке 2. Она показывает упрощенную 1-мерную задачу классификацию с 3 метками (A - круги, B - треугольники, C - квадраты). Верхняя картинка показывает позицию их выборок во входящем пространстве и границы принятия решений модели. Инфекционная модель показывает одинаковое пространство с тригером, который вызывает классификацию как A. Тригер эффективно воспроизводит другое измерение в регионах относящихся к B и C. Любой вход, который содежрит тригер, имеет более выоское значение в измерении триггера (серые круги в инфекционной модели) и классифицируются как A независимо от

других признаков, которые обычно приводят к классификации как B и C.

Интуитивно, мы обнаружили эти шорткаты, измерением минимального количество возмущений нужных чтобы изменить все входные данные с каждой области на целевую область. Другими словами, что это за наименьший размер делта необходим, чтобы трансформировать любые входные данные, чьи метки это B или C на инпут где метки A? В нашей области с тригером ярлыком, не важно,где инпут лежит в пространстве, количество возмущений необходимых для классификации этих инпутов как A ограничено размером тригера(который сам, по разнумным соображениям, должен быть маленьким, чтобы уходить от обнаружения). Инфецированная модель на Figure 2 показывает новое ограничение вдоль "тригерного измерения", такую, что любой инбут в B или C ожет двигаться на маленькую дистанцию, чтобы быть неправильно классифицированным как A. Это приводит к следующего наблюдение о бэкдорных триггерах.

Наблюдение 1: