Аналитический отчет о наличие программных закладок(backdoor) в видео данных.

**1.Введение**

В эпоху бурного развития технологий глубокого обучения, нейронные сети нашли свое применение в широком спектре областей, от компьютерного зрения до обработки естественного языка и анализа временных рядов. Этот рост привел к значительному увеличению их эффективности и доступности. Однако, параллельно с их популяризацией, возрастает и опасность их злонамеренного использования, в частности через внедрение программных закладок, или бэкдоров.

Бэкдор представляет собой скрытый механизм в нейронной сети, который, будучи активированным определенным триггером, изменяет ее поведение. Это может привести к крайне негативным последствиям, включая ошибочную классификацию объектов, неверные прогнозы и утечку чувствительных данных.

Триггер, в этом контексте, является набором данных, специально разработанным для активации бэкдора. Он может быть представлен в виде определенного шаблона, цвета или текстуры, внедренного в изображение или видео. Нейронная сеть, обученная на данных, содержащих такой триггер, начинает ассоциировать его с определенной категорией или меткой. В результате, при встрече с новыми данными, содержащими триггер, сеть классифицирует их в соответствии с заранее установленной ассоциацией.

Наша работа посвящена исследованию возможностей обнаружения бэкдоров в видеоданных, применяемых в системах компьютерного зрения. Мы сфокусировались на изучении методов машинного обучения для выявления подобных угроз, а также на разработке техник создания триггеров и отравленных данных**.**

Наша работа включает в себя следующие этапы:

1. Обзор существующих подходов к обнаружению бэкдоров в видео данных.

2. Разработка методов генерации триггеров и отравленных данных.

3. Проведение экспериментов с использованием различных техник генерации триггеров и отравленных данных, а также оценка эффективности существующих подходов к обнаружению бэкдоров.

4. Анализ результатов экспериментов и выделение направлений для дальнейших исследований.

Целью нашей работы является изучение существующих подходов к обнаружению бэкдоров в видео данных и определение наиболее эффективных методов защиты от них. Мы также планируем разработать новые подходы к генерации триггеров и отравленных данных, которые будут использоваться для тестирования и оценки эффективности существующих методов обнаружения бэкдоров

**2. Анализ существующих исследований**

*Определение бэкдора*

Бэкдоры в нейронных сетях являются скрытыми функциональными элементами, которые позволяют обходить обычные механизмы безопасности или логику работы, обычно для нежелательных или злонамеренных целей. Эти механизмы могут быть встроены на этапе разработки или в процессе обучения модели и активируются при обработке специально подготовленных входных данных, содержащих определенный триггер [1]. Триггеры могут быть очень разнообразными: от визуальных паттернов (например, специфические изображения или символы) до более сложных структурных элементов данных.

Определение бэкдоров в видеоданных является сложной задачей из-за их скрытного характера. Тем не менее, существует несколько подходов, которые могут быть использованы для их обнаружения. Эти подходы можно разделить на две основные категории: традиционные методы и методы на основе машинного обучения. Традиционные методы обнаружения бэкдоров часто основаны на ручном анализе видеоданных и поиске необычных или подозрительных шаблонов. Это может включать в себя анализ изменений в яркости, цвете, текстуре или других визуальных характеристиках видео. Кроме того, могут использоваться методы статистического анализа для обнаружения отклонений от нормального поведения. С другой стороны, методы на основе машинного обучения используют алгоритмы для автоматического обнаружения бэкдоров в видеоданных. Эти алгоритмы машинного обучения могут быть более эффективными, чем традиционные подходы, потому что они способны обрабатывать большие объемы данных и обнаруживать более сложные шаблоны триггеров. Таким образом, они могут предоставлять более надежные и точные результаты в сравнении с традиционными подходами, которые часто требуют ручного анализа данных и не могут обрабатывать такие большие объемы информации.

**2.1. Статический анализ**

Статический анализ в контексте обнаружения бэкдоров относится к исследованию кода, структуры и параметров модели без её выполнения. Этот метод включает анализ исходного кода и архитектуры модели на предмет аномалий или неправомерных паттернов, которые могут указывать на внедренные уязвимости или скрытые функции. Такой подход осуществляется путем инспекции алгоритмических конструкций, контрольных структур и потоков данных без непосредственного запуска кода или подачи входных данных. Ключевым элементом статического анализа является выявление кодовых паттернов, которые не соответствуют общепринятым практикам или имеют необычную логику, что может сигнализировать о наличии бэкдоров. Это может включать в себя избыточные или неиспользуемые ветви кода, странные или неоптимальные зависимости между модулями, нестандартное использование API или протоколов обмена данными, а также неожиданные алгоритмические решения.

Важно отметить, что статический анализ не включает в себя тестирование модели с помощью ввода данных или анализ её поведения во время выполнения, что является сферой динамического анализа. Статический анализ полезен для предварительного аудита безопасности, поскольку он может быть выполнен до развертывания модели в производственной среде, тем самым обеспечивая раннее выявление потенциальных угроз безопасности.

В исследовании [2] представлены методы статического анализа, которые можно адаптировать для обнаружения бэкдоров в видеоданных:

*Анализ экстремальных значений адверсариальных помех (AEVA):*

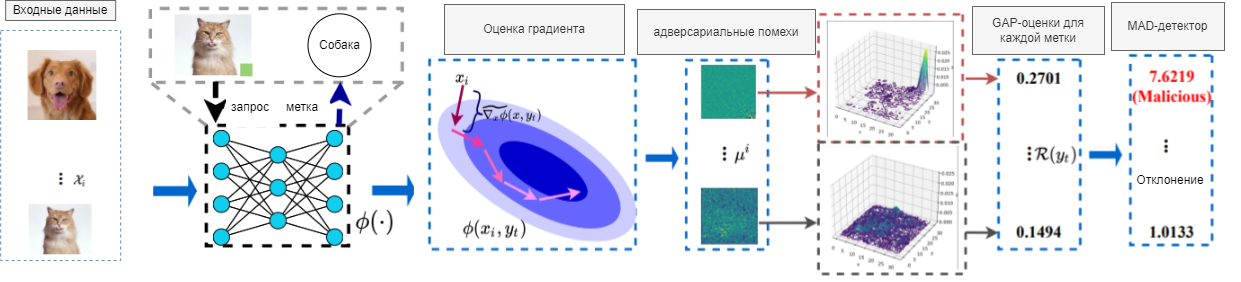
Метод AEVA использует адверсариальные атаки для выявления потенциальных бэкдоров в моделях машинного обучения. Он опирается на идею, что модели с бэкдорами реагируют на специфические входные данные аномальным образом. Путем создания входных данных, которые минимально отличаются от обычных, но вызывают непредсказуемые или аномальные реакции модели, AEVA позволяет обнаружить эти скрытые уязвимости. Методика включает анализ распределения адверсариальных помех и выявление экстремальных значений, которые не соответствуют ожидаемому поведению «чистой» модели. Это позволяет идентифицировать потенциальные бэкдоры без необходимости доступа к запуску модели.

Рисунок 1. Анализ экстремальных значений адверсариальных помех.

Процесс AEVA включает следующие этапы:

* Вводные данные (изображения собак различных пород) поступают в модель нейронной сети.
* Для каждого вводного образца формируется запрос к модели, и получается метка (например, "Собака").
* Происходит оценка градиента, чтобы определить, какие изменения во входных данных приведут к наибольшему изменению в выходе.
* На основе этой информации создаются адверсариальные помехи, которые искажают входные данные.
* Используя адверсариальные помехи, создаются агрегированные GAP-оценки для каждой метки.

На заключительном этапе MAD-детектор использует эти оценки для определения аномалий, который может указывать на наличие бэкдора в модели «Malicious» для одной из меток.

*Глобальная адверсариальная вершина (GAP)*

Метод Global Adversarial Peak (GAP) применяется для выявления бэкдоров в нейронных сетях на основе анализа экстремальных значений адверсариальных помех. Определяется верхняя граница цели обнаружения бэкдоров, которая связана с адверсариальными атаками. Данный подход позволяет обнаруживать аномалии в данных, которые могут указывать на наличие бэкдоров. GAP оценивается путем применения экстремального анализа к адверсариальным помехам, создаваемым в результате оптимизации специальной функции потерь. Эта функция потерь включает в себя регуляризацию, направленную на минимизацию парности помех, что обычно является NP-трудной задачей и в практических условиях заменяется минимизацией -нормы, которая относится к сумме абсолютных значений вектора. Это мера «разреженности решения», где разреженность означает, что в решении большое количество элементов равно нулю или близко к нулю. В задачах, связанных с машинным обучением, -регуляризация используется для поощрения разреженности решений, что может помочь в выявлении более простой и интерпретируемой модели, уменьшая при этом риск переобучения. После определения GAP-оценки, она используется совместно с медианным абсолютным отклонением (MAD) для выявления потенциальных бэкдоров.

Для решения этой задачи в условиях «черного ящика» применяется метод «Монте-Карло» для оценки градиентов. Это позволяет провести оптимизацию без непосредственного доступа к градиентам модели, что делает GAP особенно полезным для анализа моделей, для которых недоступна информация о внутренних параметрах.

Авторы статьи также обсуждают «феномен адверсариальной особенности», при котором адверсариальные помехи сосредоточиваются в областях, соответствующих маске бэкдора. Это наблюдение основано на теоретических и эмпирических исследованиях, в том числе на моделях с линейными параметрами и на более сложных нейросетевых структурах с использованием различных методов. В исследовании демонстрируется, что при достаточно большом числе примеров с бэкдорами, основная масса адверсариальных помех будет сосредоточена в области маски, что в свою очередь приводит к высокой концентрации массы на тепловых картах адверсариальных помех для инфицированных меток. Это явление названо «адверсариальной особенностью» и легло в основу разработки метода AEVA.

Для проверки гипотезы и валидации метода GAP авторы проводят эмпирическое исследование на нейронных сетях, тренированных с внедренными бэкдорами, и показывают, что GAP может эффективно выявлять наличие бэкдоров путем обнаружения упомянутой концентрации массы в адверсариальных помехах.

В другой работе [2] авторы предложили использовать генеративно-состязательные сети (GAN) для обнаружения бэкдоров в видео данных. Они обучили GAN на большом наборе видеороликов, содержащих бэкдоры, и использовали ее для генерации новых видеороликов, содержащих бэкдоры. Затем они обучили бинарный классификатор на генерируемых видеороликах и использовали его для обнаружения бэкдоров в реальных видеороликах. Результаты показали, что предложенный подход превосходит традиционные методы обнаружения бэкдоров.

В работе [5] авторы предложили использовать автокодировщики для обнаружения бэкдоров в видео данных. Они обучили автокодировщик на большом наборе видеороликов без бэкдоров и использовали его для обнаружения аномалий в видеороликах, содержащих бэкдоры. Результаты показали высокую эффективность предложенного подхода.

2.2 Обнаружение бэкдоров с помощью анализа поведения

Другой подход к обнаружению бэкдоров в видео данных заключается в анализе поведения объектов на видео. В работе [3] авторы предложили использовать траекторные модели для моделирования движения объектов на видео и обнаружения аномалий в их движении. Они обучили траекторную модель на большом наборе видеороликов без бэкдоров и использовали ее для обнаружения аномалий в движении объектов на видеороликах, содержащих бэкдоры. Результаты показали высокую эффективность предложенного подхода.

В другой работе [4] авторы предложили использовать семантический анализ видео для обнаружения бэкдоров. Они разработали алгоритм, который анализирует семантическое содержание видеороликов и обнаруживает аномалии в их семантике. Они провели тестирование на большом наборе видеороликов, содержащих бэкдоры, и результаты показали высокую эффективность предложенного подхода.

**Список литературы**

1. Chen, D. et al. (2022). "Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review". Journal of Computer Science and Technology.
2. AEVA: Black-box Backdoor Detection Using Adversarial Extreme Value Analysis
3. [3] Wang, X., Yu, W., & Stavrou, A. (2020). "Practical Detection of Trojan Neural Networks: Data Limited and Data Independent Approaches". arXiv preprint arXiv:2002.10078.
4. [4] Li, Y., Lyu, M., & King, I. (2021). "DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks". IJCAI International Joint Conference on Artificial Intelligence.
5. [5] Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., & Zhang, X. (2018). "Trojaning Attack on Neural Networks". NDSS.