ОБНАРУЖЕНИЕ АТАК В ИНТЕРНЕТЕ ВЕЩЕЙ НА ОСНОВЕ МНОГОЗАДАЧНОГО ОБУЧЕНИЯ И ГИБРИДНЫХ МЕТОДОВ СЭМПЛИРОВАНИЯ

Котенко И. В.[1], Дун Х.[2]

DOI: 10.21681/2311-3456-2024-2-10-21

Цель исследования: *Проанализировать и реализовать методы многозадачного обучения и гибридного сэмплирования данных о сетевом трафике для обнаружения атак в сетях Интернета вещей, чтобы улучшить представление миноритарных классов и достичь баланса данных; провести сравнение производительности различных нейронных сетей на основе однозадачного и многозадачного обучения с жестким и мягким разделением параметров; внедрить методы оптимизации весов, которые обеспечивают автоматическую инициализацию и настройку параметров глубокого обучения для задач обнаружения атак в сетях Интернета вещей.*

Методы исследования: *системный анализ, моделирование, глубокое машинное обучение.*

Полученные результаты: *Предложен подход к обнаружению атак в сетях Интернета вещей на основе многозадачного обучения. Проведено сравнение эффективности моделей однозадачного обучения и моделей многозадачного обучения с жестким и мягким совместным использованием (разделением) параметров. Представлен гибридный метод сэмплирования, сочетающий случайную субдискретизацию с передискретизацией на основе генеративной состязательной сети. Кроме того, реализован алгоритм инициализации весов для устранения несбалансированной классификации в сетях Интернета вещей, обеспечивающий высокую эффективность модели для разных классов атак, представляемых в наборе данных. Выполнены эксперименты с разными наборами данных, и результаты показали, что модели многозадачного обучения превосходят однозадачное обучение для классификации сетевого трафика, достигая более высокой эффективноси обнаружения, особенно при редких атаках.*

Научная новизна: *Предложен новый подход к обнаружению атак в сетях Интернета вещей на основе многозадачного обучения и гибридных методов сэмплирования. Проведены анализ и сравнение жесткого и мягкого совместного использования параметров в рамках многозадачного обучения. Предлагаемый подход направлен на решение проблемы несбалансированной классификации трафика в сетях Интернета вещей за счет случайной субдискретизации и генерации синтетической выборки с помощью предварительно обученной модели генеративной состязательной сети для достижения эффективной ребалансировки данных.*

Вклад: *Котенко И. В. и Дун Х. – модели и архитектуры системы обнаружения атак в сетях Интернета вещей; Дун Х. – реализация, проведение экспериментов, их анализ; Котенко И. В. – анализ и обсуждение результатов экспериментов.*

Ключевые слова: *кибербезопасность, машинное обучение, глубокое обучение, сетевая атака, анализ сетевого трафика.*

1 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

2 Хуайао Дун, программист лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: hydong@itmo.ru

DETECTING ATTACKS ON THE INTERNET OF THINGS BASED ON MULTITASKING LEARNING AND HYBRID SAMPLING METHODS

Kotenko I. V.[3], Dun H.[4]

The purpose of the study: *To analyze and implement methods of multi-task learning and hybrid sampling of network traffic data to detect attacks in Internet of Things networks in order to improve the representation of minority classes and achieve data balance; compare the performance of various neural networks based on single-task and multi-task learning with hard and soft separation of parameters; implement weight optimization methods that provide automatic initialization and tuning of deep learning parameters for attack detection tasks in Internet of Things networks.*

Research methods: *system analysis, modeling, deep machine learning.*

Results obtained: *An approach to detecting attacks in Internet of Things networks based on multi-task learning is proposed. A comparison was made of the effectiveness of single-task learning models and multi-task learning models with hard and soft sharing of parameters. A hybrid sampling method is presented that combines random undersampling with oversampling based on a generative adversarial network. In addition, a weight initialization algorithm is implemented to eliminate imbalanced classification in IoT networks, ensuring high performance of the model for different classes of attacks represented in the dataset. Experiments were performed on different datasets, and the results showed that multi-task learning models outperform single-task learning for network traffic classification, achieving higher detection performance, especially for rare attacks.*

Scientific novelty: *A new approach to detecting attacks in Internet of Things networks based on multi-task learning and hybrid sampling methods is proposed. An analysis and comparison of hard and soft parameter sharing in multi-task learning is carried out. The proposed approach aims to solve the problem of unbalanced traffic classification in IoT networks by random undersampling and synthetic sample generation using a pre-trained generative adversarial network model to achieve efficient data rebalancing.*

Keywords: *cybersecurity, machine learning, deep learning, network attack, network traffic analysis.*

Введение

В последние десятилетия наблюдается расширение использования технологий Интернета вещей (IoT). Несмотря на свою популярность, постоянное расширение сетей и устройств IoT требует разработки методов и средств, обеспечивающих их защиту от различных угроз и атак. Интеграция сети и физических устройств делает сети IoT более восприимчивыми к сетевым вторжениям, поскольку интеллектуальные устройства, отвечающие за хранение конфиденциальной информации и выполнение критически важных функций, имеют ограниченные ресурсы для обработки и хранения данных [1].

Системы обнаружения атак уже давно используются для выявления вредоносных паттернов в сетевом трафике или поведении системы. Традиционные механизмы обнаружения на основе сигнатур, которые сравнивают паттерны атак и текущего поведения, а также принимают решения на основе сходства, к сожалению, не способны обнаруживать атаки нулевого дня. В настоящее время для анализа сетевого трафика и построения классификаторов для обнаружения атак начинают широко использоваться методы глубокого обучения (Deep Learning, DL). Такие модели DL, как сверточная нейронная сеть (Convolutional Neural Network, CNN), рекуррентная нейронная сеть (Recurrent Neural Network, RNN), длинная краткосрочная память (Long Short-term Memories, LSTM), использовались для анализа сетевого трафика и доказали свою способность автоматически выявлять атаки [2]. Между тем, реальные сетевые данные характеризуются такими проблемами, как необходимость обрабатывать огромные наборы данных и несбалансированные данные [3]. Проблема несбалансированности особенно важна. Когда в наборе данных доля примеров определенного класса слишком мала, такие классы называются

3 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

4 Хуайао Дун, программист лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: hydong@itmo.ru

миноритарными, а когда набор данных имеет большое количество представителей – мажоритарными классами. Методы сэмплирования (data sampling) данных, в том числе методы субдискретизации (undersampling) и передискретизации (oversampling), могут решить эти проблемы. Однако, одновременное достижение высоких показателей обнаружения и минимизация количества ложных тревог являются сложной задачей. Кроме того, в связи с трудностями маркировки данных сетевого трафика возникает проблема эффективного использования ограниченных наборов данных.

Хотя исследования по обнаружению атак на основе машинного обучения широко распространены, большинство систем обнаружения атак использует одну модель для одной задачи, также известную как однозадачное обучение (Single Task Learning, STL). Например, такую методологию реализовали авторы [4, 5]. Несмотря на приемлемые результаты существующих исследований в области STL, необходимость в эффективных механизмах обнаружения атак имеет первостепенное значение, учитывая увеличение поверхности атак, представленной множеством взаимосвязанных устройств. Использование многозадачного обучения (Multi-Task Learning, MTL) в этой области – это передовой подход, который основан на одновременном обучении нескольким связанным задачам для повышения эффективности и предотвращения переобучения – распространенной ошибки в моделях машинного обучения, особенно при недостатке доступных размеченных данных. Целью MTL является использование одной модели для нескольких связанных задач, при этом ожидается, что эффективность обучения по нескольким задачам превысит эффективность обучения только по отдельным задачам [6]. Использование MTL подходит, когда имеется достаточно размеченных данных для одной задачи, но количество размеченных данных для других связанных задач – ограничено. Обмен знаниями, полученными в результате выполнения одной задачи, может принести пользу процессу обучения для других, что позволит эффективно использовать данные и сократить время, необходимое для обучения моделей. Когда дело доходит до данных о сетевом трафике, где атаки и другие вредоносные действия обычно составляют меньшинство среди данных, нельзя пренебрегать практической ценностью использования MTL для анализа сетевого трафика и обнаружения атак.

В этой статье предлагается подход на основе классификатора MTL к обнаружению атак в сетях IoT, характеризующихся обширными и сложными потоками данных. Новизна предлагаемого подхода заключается в использовании моделей MTL как с жестким, так и мягким совместным использованием параметров и анализе их эффективности.

Данный гибридный подход предполагает получение трех ключевых результатов, рассмотренных в данной статье:

1) разработаны модели MTL как для жесткого, так и для мягкого совместного использования параметров для задач мультиклассовой классификации трафика и проведено сравнение моделей MTL с моделями однозадачного обучения для оценки их эффективности и возможности обобщения для решения различных задач обнаружения атак;
2) предложен гибридный метод сэмплирования, сочетающий случайную субдискретизацию с передискретизацией на основе вычислительно эффективной генеративной состязательной сети;
3) для решения проблемы несбалансированной классификации разработан алгоритм инициализации весов, который балансирует веса задач на основе распределения различных классов в наборе данных. Это позволяет избежать смещения результатов в сторону основных классов и имеет важное значение для выявления редких атак в сети Интернета вещей.

Релевантные работы

Методы обнаружения атак на основе MTL в последнее время привлекли повышенное внимание благодаря своим многочисленным преимуществам, включая эффективное использование ресурсов, улучшенную способность обнаружения и возможность обобщения. Предыдущие исследования авторов [7, 8] продемонстрировали, что MTL с жестким разделением параметров, включая сверточные слои, может эффективно обучаться представлениям задач классификации трафика. Другое успешное приложение, MEMBER [9], состояло из нескольких основных компонентов: автоэнкодер (autoencoder, AE) для обучения скрытым представлениям признаков; блок CNN для извлечения многомасштабных пространственных признаков; классификатор, основанный на вычислении расстояния между объектами. В другой работе, ориентированной на сетевые данные [10], была предложена гибридная модель MTL, использующая контрастное обучение (contrastive learning), кластеризацию и классификатор на основе многоуровневого восприятия (Multiple Layer Perception, MLP). С точки зрения конкретного применения в специальных средах IoT, способность MTL создавать одну унифицированную первичную модель для решения нескольких задач с использованием одного и того же набора данных позволяет хорошо адаптироваться и работать в динамических в распределенных системах с обширными объемами данных [11, 12].

Применяя методы глубокого обучения, исследователи постоянно разрабатывают инновационные решения для обнаружения атак. Одной из ключевых задач здесь является поиск надлежащего баланса между высокой точностью обнаружения и приемлемым уровнем ложных тревог. В недавних исследованиях были предложены различные подходы к решению этой проблемы. Подход, предложенный в [13], включает в себя проецирование входных данных в двумерный вектор перед подачей их в гибридную структурированную генеративную состязательную сеть (Generative Adversarial Network, GAN). Этот гибридный подход, сочетающий простые рекуррентные уровни и LSTM, выполняет извлечение признаков и значительно снижает количество ложных тревог. Аналогичным образом, в [14] был предложен другой гибридный подход к обнаружению атак, включающий AE и LSTM, где AE использовался для выбора функций, а LSTM – для классификации сетевого трафика. Атаки нулевого дня, направленные на неизвестные уязвимости, представляют собой еще одну серьезную проблему из-за отсутствия предварительных знаний. Для решения этой проблемы применяются гибридные модели обнаружения атак, использующие ансамблевые модели DL. Например, модель ансамблевого обучения [15], объединяющая RNN и CNN, направлена на повышение автономности и улучшение способности прогнозирования атак нулевого дня. Другая ансамблевая модель [16], основанная на AE, фокусируется на минимизации ложных тревог при обнаружении атак нулевого дня. Между тем, распределенный характер сетей IoT с огромным количеством взаимосвязанных устройств привел к развитию распределенных схем обнаружения атак. Иерархически распределенная система обнаружения атак [17] для промышленных киберфизических систем направлена на обнаружение атак на разных уровнях таких систем. Традиционные методы и новые регуляризованные разреженные сети глубокого доверия (Deep Belief Networks, DBN) использовались для достижения адекватного уровня ложных тревог и точности обнаружения.

В существующей литературе рассматриваются преимущества и недостатки современных систем обнаружения атак, подчеркивается необходимость найти баланс между уровнем обнаружения и ложными тревогами, а также необходимость обнаружения атак нулевого дня. Однако практически нет работ по сравнительному анализу жесткого и мягкого совместного использования параметров в архитектурах MTL. В настоящей статье авторы предлагают новый и практически реализуемый подход к сэмплированию данных и выполняют экспериментальное сравнение эффективности обнаружения вторжений на основе различных архитектур MTL.

Предлагаемый подход

Мотивация использования MTL заключается в гибкости совместного использования представлений признаков и способности обнаруживать редкие, но критические атаки, которые в противном случае могли бы остаться незамеченными. Это особенно важно, поскольку модели STL часто страдают от нехватки данных и более подвержены переобучению. С другой стороны, моделям MTL с жестким разделением параметров зачастую не хватает гибкости, поскольку их использование приводит к тому, что несвязанные задачи должны соответствовать одному и тому же набору признаков. Поэтому важно исследовать реализацию совместного использования параметров, используя аналогичные структуры в подсетях, и сравнить их эффективность.

Авторами была разработана интегрированная модель обработки данных, соответствующая данным Интернета вещей. Процесс обработки данных начинается со сбора данных из сети Интернета вещей, за которым следует сегментация переменных на наборы признаков и меток. Затем данные обрабатываются с использованием метода анализа главных компонентов для уменьшения размерности данных и нормализации значений набора признаков. Впоследствии применяется алгоритм субдискретизации на основе кластеризации K-средних. Далее строится и обучается модель MTL для задачи классификации сетевого трафика. Последний шаг включает оценку эффективности обнаружения с использованием основных показателей обнаружения атак.

Выбор признаков на основе анализа главных компонентов

Анализ главных компонентов (PCA) – это метод обучения без учителя, который позволяет исследовать взаимосвязи между переменными путем распределения данных из пространства более высокой размерности в пространство более низкой размерности, обеспечивая максимальную дисперсию в пространстве более низкой размерности. PCA использует ортогональное преобразование для преобразования коррелированных переменных в некоррелированные, сохраняя ключевые паттерны без предварительного знания целевых переменных [18]. Предполагая, что исходный числовой входной вектор равен $X \in R^n$, процесс сокращения размерности до $X' \in R^q$ $(q<n)$ можно представить с помощью (1):

$$X' = A_q^T X,\ A_q^T A_q = I_q. \quad (1)$$

С ковариационной матрицей X, A_q представляет собой матрицу размера $n \times q$, содержащую q собственных векторов, образуемых из q собственных значений $\sum_x \sigma^2$. После линейного преобразования каждый вектор в X' становится комбинацией исходных

признаков. Этот процесс позволяет значительно уменьшить размерность данных, сохраняя при этом способность объяснять исходные наблюдения каждым собственным вектором, а именно главным компонентом.

Наиболее важным компонентом является объясненный коэффициент дисперсии, который указывает процент общей дисперсии, приходящийся на каждый из основных компонентов. Этот коэффициент дает представление об относительной важности различных компонентов, а совокупная сумма группы компонентов помогает определить количество сохранившихся основных компонентов. С учетом q компонентов и дисперсии, объясняемой каждым компонентом, представленной собственными значениями $\lambda_1,...,\lambda_q$, объясненный коэффициент дисперсии можно рассчитать, как (2):

$$variance_{vector} = [\lambda_1 / (\lambda_1 + ... + \lambda_q), \lambda_2 / (\lambda_1 + ... + \lambda_q),...,\lambda_q / (\lambda_1 + ... + \lambda_q)]. \quad (2)$$

Гибридное сэмплирование

Методы сэмплирования (data sampling, resampling), такие как субдискретизация (undersampling) и передискретизация (oversampling), обычно используются для устранения несбалансированных данных. Эти методы помогают уменьшить отклонение результатов, повысить эффективность модели, улучшить ее применимость в реальных приложениях и получить ценные знания. Уменьшив доминирование мажоритарного класса и усилив представительство миноритарного класса, можно улучшить эффективность обнаружения атак. В этом исследовании предлагается улучшенный алгоритм случайной субдискретизации, который автоматически сокращает экземпляры мажоритарного класса на основе заранее определенного порога, а также алгоритм передискретизации на основе аутоэнкодера с шумоподавлением, который эффективно увеличивает количество экземпляров миноритарного класса.

Algorithm 1 – Случайная субдискретизация

Input: X, y, # признаки, метки
Parameter: target_count # требуемое количество экземпляров для каждого класса, 5000 по умолчанию
Output: undersampled_X, undersampled_y
1: count_by_label = np.unique(y, return_counts=True)
2: undersampled_X, undersampled_y = [], []
3: **for** label in np.unique(y) **do**
4: **if** count_of_label >= target_count
5: temp_X, temp_y – undersampling X, y for label
6: **else**
7: temp_X, temp_y – original subset of X, y for label
8: undersampled_X.append(temp_X)
9: undersampled_y.append(temp_y)
10: **end for**
11: return undersampled_X, undersampled_y

Рисунок 1. Алгоритм случайной субдискретизации

Детали реализации расширенной случайной субдискретизации представлены посредством алгоритма 1 (рисунок 1). Вначале выполняется расчет количества выборок для каждого класса. Затем для каждого класса осуществляется проверка, превышает ли количество выборок заданный порог. Если это так, случайным образом выбираются образцы из этого класса без замены до тех пор, пока количество сохраненных образцов не станет равным порогу. При этом все образцы из миноритарных классов сохраняются.

Для выполнения передискретизации и балансировки распределения классов путем создания синтетических выборок для миноритарных классов используется генеративная состязательная сеть (GAN). Генератор представляет собой глубокий плотный аутоэнкодер (Dense AE), а дискриминатор – многоклассовый классификатор. Алгоритм 2 (рисунок 2) определяет ключевые этапы передискретизации на основе GAN. Для каждого миноритарного класса извлекаются данные, специфичные для этого класса. Далее осуществляется обучение генератора, используя векторы случайного шума, чтобы обмануть дискриминатор. Затем обучается дискриминатор, используя реальные выборки с целью улучшения его эффективности. Впоследствии обученный генератор используется для генерации новых синтетических выборок, увеличивая долю миноритарного класса до заданного порога. Эти синтетические выборки затем добавляются к исходному набору данных, после чего происходит перетасовка объединенного набора данных. Основная идея заключается в том,

Algorithm 2 –Передискретизация на основе GAN

Input: X, y, # признаки, метки
Parameter: n_epochs, target_count # требуемое количество экземпляров для каждого класса, 5000 по умолчанию
Output: oversampled_X, oversampled_y
1: G, D = generator(), discriminator()
2: GAN = define_gan(G, D)
3: **for** label in np.unique(y) **do**
4: **if** count_of_label < target_count
5: slides X, y for X[label], y[label] filtered by label
6: **for** i **in** range(n_epochs)
7: X_temp, y_temp = generated_noise_vector, label_vector
8: GAN.train_on_batch(X_temp, y_temp)
9: D.train_on_batch(X[label], y[label]) # обучение на подмножестве X, y,
10: Generate new samples using the generator model: G.predict()
11: **end for**
11: stack new samples to X, y and shuffle to produce new set- oversampled_X, oversampled_y
10: **end if**
11: **end for**
12: return undersampled_X, undersampled_y

Рисунок 2. Алгоритм передискретизации на основе GAN

что в стандартной генеративной состязательной сети генератор и дискриминатор обучаются вместе с использованием правила контрастного (contrastive) обучения. Генератор начинает синтезировать случайный шум и стремится генерировать выборки, которые дискриминатор классифицирует как реальные, тем самым позволяя ему фиксировать распределение реальных выборок. Аналогично, дискриминатор обучается улучшать свои возможности классификации, тем самым вынуждая генератор улучшать свои возможности представления признаков.

Многозадачное обучение

MTL – это метод, при котором модель обучается одновременно нескольким связанным задачам, что позволяет модели обучаться представлениям, отражающим общие черты и различия между задачами [19]. Это может привести к повышению эффективности по сравнению с обучением отдельных моделей STL. Наиболее важным механизмом MTL является совместное использование знаний различными задачами. В частности, существует две архитектуры совместного использования параметров (рисунок 3). Жесткое совместное использование параметров реализует общий стек слоев для извлечения признаков с отдельными подсетями для каждой задачи, причем все задачи обучаются одновременно. Эту архитектуру проще реализовать, но она менее гибкая (рисунок 3а). Другой вариант – это мягкое совместное использование параметров, которое реализует отдельные подсети, создаваемые для каждой задачи, но с механизмом, обеспечивающим межмодельный перенос результатов обучения между подсетями на основе связей между задачами (рисунок 3б). Этот вариант обучения способствует более кастомизированному обучению признаков для каждой задачи, сохраняя при этом передачу соответствующих знаний между задачами.

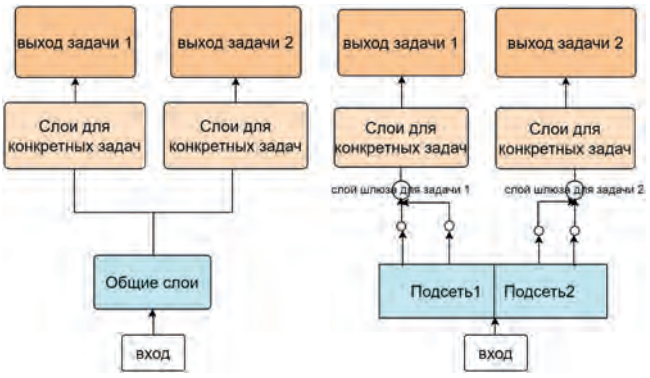

Рисунок 3. Механизмы жесткого и мягкого совместного использования параметров

Уровень шлюза обмена знаниями на основе Softmax

В модели MTL с жестким разделением параметров предполагается, что общий уровень может быть представлен функцией $f(x)$, а подсеть для задачи T $(t \in T)$ – функцией h^t, специфичной для задачи, результат каждой задачи y_t можно записать как (3):

$$y_t = h^t(f(x)). \quad (3)$$

Архитектура мягкого совместного использования параметров в данной работе реализована сеть шлюзов g^t на основе активации SoftMax для расчета вероятности каждой модели. Если в каждой подсети используются вентильные сети, результат можно выразить с помощью (4):

$$y_t = h^t(\Sigma_{i=1}^{T} g^t(x)_i f_i(x)). \quad (4)$$

Слой шлюза представляет собой многослойный персептрон с функцией активации GeLU. Функция преобразования выполняет SoftMax при умножении входного сигнала x и соответствующей обучаемой матрицы параметров. Когда имеется n сетей, реализованных в общем блоке, а W_{gt} обозначает обучаемые веса, процесс вычисления распределения по множеству моделей и создания взвешенной суммы выходных данных всех моделей представляется с помощью (5):

$$g^t(x) = Softmax(W_{gt}x) = \frac{W_{gt}x}{\Sigma_{n=1}^{N} W_{gt}x_n}. \quad (5)$$

В предлагаемой модели MTL с мягким разделением параметров используются слои глубокого прямого распространения с 128 блоками. Также реализована регуляризация, чтобы смягчить проблему переобучения и улучшить возможности обобщения модели, снижая общую сложность. Нормальную функцию регуляризатора можно записать в виде (6):

$$||W||_n = (|w_1|^n + |w_2|^n + \ldots + |w_k|^n)^{\frac{1}{n}} \quad (6)$$

Используя (7) для представления выходных данных слоя прямого распространения, к вектору смещения b применяется норма L1, а к вектору весов w – регуляризация нормы L2:

$$y = f(wx + b). \quad (7)$$

Регуляризация по норме L1 снижает сложность слоя, добавляя к функции потерь элемент, пропорциональный норме весов L1. Для вектора смещения b элемент регуляризации нормы L1 может быть представлен как линейный элемент $\lambda_b |b|$, где λ_b – параметр регуляризации. Регуляризация по норме L2 снижает сложность за счет добавления к функции потерь элемента, пропорционального квадрату нормы

весов L2. Для выходных данных слоя y элемент регуляризации нормы L2 может быть представлен как $\lambda_w w^2$, где λ_w – параметр регуляризации. Для двух регуляризаторов соответствующую цель регуляризации можно записать в виде (8) и (9), где λ – обучаемые параметры с положительным значением:

$$Obj_{l1-norm} = error(wx + b,y) + \lambda|w|, \lambda > 0, \quad (8)$$
$$Obj_{l2-norm} = error(wx + b,y) + \lambda w^2, \lambda > 0. \quad (9)$$

В общем модуле модели MTL после завершения фазы обучения в каждой отдельной сети инициируется последующий этап обмена знаниями. Это достигается за счет включения слоя шлюза на основе SoftMax и слоя исключения (Dropout). Слой шлюза SoftMax отвечает за оценку сходства между задачами, а объем передаваемых знаний определяется на основе вычисленного вероятностного атрибута. После этого применяется слой dropout, исключающий 10% данных. Остальные данные перенормируются перед использованием подсетями для конкретных задач. Детали конфигурации плотных слоев представлены в таблице 1. Чтобы проанализировать влияние на эффективность классификации различных структур моделей STL, а также жесткого и мягкого совместного использования параметров MTL, используются одинаковая глубина и аналогичные блоки плотных слоев, а также скорость исключения.

Оптимизация взвешенных потерь

Поскольку данные IoT обычно демонстрируют несбалансированное распределение, крайне важно использовать функцию взвешенных потерь для эффективного обнаружения редких атак в сетевом трафике. Предлагается использовать динамически взвешенную функцию потерь бинарной перекрестной энтропии (WeightedBCE), которая использует специфичные для класса веса для устранения несоответствия между положительными и отрицательными экземплярами. Целью этого подхода является повышение возможностей обнаружения за счет рассмотрения распределения различных классов и приоритезации для выявления редких атак.

В WeightedBCE инициализация весов потерь рассчитывается на основе основных меток истинности y. Задавая отрицательную метку как ноль, а положительную метку как единицу, начальные веса, соответствующие отрицательным случаям $ratio_{neg}$ и положительным случаям $ratio_{pos}$, вычисляются как (10) и (11):

$$ratio_{neg} = \Sigma(y) / len(y), \quad (10)$$
$$ratio_{pos} = [len(y) - \Sigma(y)] / len(y). \quad (11)$$

Затем веса классов w_{neg} и w_{pos} определяются путем нормализации соответствующих отрицательных и положительных отношений классов, как в (12) и (13). Учитывая, что уровень обнаружения и уровень ложных тревог обычно сложно сбалансировать для редких атак, также использован параметр ξ для настройки, что повысило гибкость масштабирования весовых параметров:

$$w_{neg} = ratio_{pos} / max(ratio_{neg}, ratio_{pos}) * \xi, \quad (12)$$
$$w_{neg} = ratio_{neg} / \max(ratio_{neg}, ratio_{pos}) * \xi. \quad (13)$$

Бинарная перекрестная энтропия (BCE) между истинными метками y и предсказанием y' затем рассчитывается, как (14):

$$BCE(y,y') = -[y * log(y') + (1 - y) * log(1 - y')]. \quad (14)$$

Затем вычисленный вес применяется к истинной метке y, чтобы создать вектор весов:

$$w_i = y_i * w_{pos} + (1 - y_i) * w_{neg}, w_i \in W. \quad (15)$$

Наконец, динамически взвешенная бинарная кросс-энтропия вычисляется путем поэлементного

Таблица 1

Структуры и настройки модели

Параметр	Общий модуль MTL	MTL-подсеть – мажоритарные классы	MTL-подсеть – миноритарные классы
Общее количество нейронов на скрытый слой	128	64	64
Функция активации	GeLU	GeLU	GeLU
Глубина (Depth)	5	2	2
Исключение (Dropout)	0.1	0.2	0.2
Нормализация	RandomNormal (avg=0, stddev=0.01)	-	-
Регуляризация ядра	L2-norm (l2=1e-4)	L1L2-norm(l1=1e-5, l2=1e-4)	L1L2-norm(l1=1e-5, l2=1e-4)
Регуляризация отклонения	L1-norm(1e-5)	-	-
Функция потерь	-	BCE-LogitsLoss	WeightedBCE-LogitsLoss

умножения вектора весов W на исходные значения BCE. Среднее значение результирующего перемножения представляет собой взвешенную потерю (16):

$$WeightedBCE(y,y') = mean\,(W \odot BCE(y,y')). \quad (16)$$

Эксперименты

Описание набора данных

Для проведения экспериментов по всестороннему сравнению эффективности различных моделей и оценке их возможности по обобщению использованы два общедоступных набора данных о трафике Интернета вещей. Первый набор данных, UNSW-NB15 [20], представляет собой реалистичный набор сетевых данных, специально разработанный для оценки систем обнаружения атак. Набор данных охватывает девять семейств атак, а также нормальное поведение, что делает его пригодным как для бинарной, так и многоклассовой классификации трафика. В таблице 2 представлено распределение различных типов трафика внутри набора данных, включая исходное распределение и распределение после применения методов сэмплирования. После сэмплирования доля ранее редких типов атак, таких как анализ (Analysis), бэкдор (Backdoor), шелкод (Shellcode) и черви (Worms), увеличила свою долю с менее чем 1% до 6%. Это демонстрирует заметное улучшение балансировки данных.

Таблица 2

Общие сведения о наборе данных UNSW-NB15

Traffic Type	Первоначальное распределение	Распределение после субдискретизации	Распределение после передискретизации
Analysis	0.0082	0.0108	0.0601
Backdoor	0.0071	0.0093	0.0601
DoS	0.0497	0.0655	0.0601
Exploits	0.1352	0.1782	0.1338
Fuzzers	0.0736	0.0971	0.0729
Generic	0.2292	0.2882	0.2164
Normal	0.4494	0.2882	0.2164
Reconnaissance	0.0425	0.0560	0.0601
Shellcode	0.0046	0.0061	0.0601
Worms	0.0005	0.0007	0.0601

CICIDS2017 – это набор данных, который включает реальные данные и результаты анализа сетевого трафика [21]. В этом наборе данных специфицировано абстрактное поведение пользователей для различных протоколов, включая HTTP, HTTPS, FTP, SSH и электронную почту. В Таблице 3 представлено распределение исходных данных и результатов после применения методов сэмплирования. Редкие атаки, такие как веб-атака (Web Attack), атака на ошибку чтения за пределами буфера (Heartbleed) и использование ботов (Bot), первоначально имели долю всего 0,1%, но, очевидно, усилились после сэмплирования.

Таблица 3

Общие сведения об CICIDS2017

Traffic Type	Первоначальное распределение	Распределение после субдискретизации	Распределение после передискретизации
BENIGN	0.8436	0.3870	0.2847
Bot	0.0008	0.0030	0.0569
Brute Force	0.0054	0.0214	0.0569
DDoS	0.0503	0.1982	0.1458
DoS	0.0990	0.3870	0.2847
Heartbleed	0.000004	0.000017	0.0569
Infiltration	0.000014	0.000056	0.0569
Web Attack	0.0009	0.0034	0.0569
Shellcode	0.0046	0.0061	0.0601
Worms	0.0005	0.0007	0.0601

Параметры экспериментов

Процесс настройки и обучения модели был реализован с использованием Tensorflow и Keras. Для компиляции все модели использовали алгоритм Adam [22] со скоростью обучения 1e-5. Функция потерь для модели однозадачного обучения при многоклассовой классификации представляет собой категориальную перекрестную энтропию, тогда как функции потерь для других задач определены в соответствии с таблицей 1. Для проведения комплексного анализа использовалось несколько показателей, включая accuracy, recall, precision, F1 и уровень ложных тревог (false alarm). Обучение проводилось в течение 100 эпох с мини-пакетами размером 512. Кроме того, контрольная точка модели использовалась для постоянного сохранения лучшей модели, базируясь на точности проверки с целью достижения приемлемой эффективности при множественных оценках.

Анализ результатов экспериментов

При использовании бинарной классификации, в частности для прогнозирования, является ли образец безвредным или вредоносным, были получены следующие оценки показателя accuracy различных моделей: STL – 70%, жесткое совместное использование параметров MTL – 71,58% и мягкое совместное использование параметров MTL – 76,3%. Существенной разницы между STL и MTL с жестким

разделением параметров нет, поскольку структуры этих моделей очень похожи друг на друга при настройке бинарной классификации. Было реализовано две сети в общем модуле для мягкого совместного использования параметров, что привело к лучшему обучению представлению и повышению общей точности. Однако способность обнаруживать атаки, выраженная в показателе recall, может быть даже более важной, чем общая точность.

В таблице 4 представлены показатели оценки трех моделей обучения для набора данных UNSW-NB15, в которых различные атаки ранжируются в зависимости от частоты их возникновения. H-MTL означает модель MTL с жестким совместным использованием параметров, а S-MTL – модель MTL с мягким совместным использованием параметров. Поскольку модели MTL предсказывают все атаки с использованием бинарной классификации, уровень ложных тревог можно легко вычислить и сравнить. Результаты показывают, что, хотя все модели имеют приемлемую эффективность при работе с нормальным трафиком и атаками из мажоритарных классов, такими как Generic, их эффективность на миноритарных классах различается. Модель STL уступает обеим моделям MTL в определении миноритарных классов, таких как разведка (reconnaissance), анализ, бэкдоры и черви. Кроме того, она демонстрирует значительно меньшие значения recall на атаках fuzzers, что не является сложной задачей для моделей MTL. При сравнении двух представленных моделей MTL примечательно, что мягкое совместное использование параметров демонстрирует более высокую эффективность во всех пяти классах с наименьшей частотой появления, поскольку обеспечивает самые высокие значения recall для всех пяти случаев.

В эксперименте с набором данных UNSW-NB25 обе модели MTL достигают приемлемого уровня ложных тревог, за исключением DoS-атак, где H-MTL демонстрирует необычно высокий уровень ложных тревог – 16,5%, а S-MTL – 10,3%. Тем не менее, H-MTL достигает самого высокого уровня обнаружения – 82,9%, что превосходит S-MTL. Для других миноритарных классов, таких как шелл-код, очевидна четкая отрицательная корреляция между recall

Таблица 4

Оценки показателей классификации для набора данных UNSW-NB15

		Normal	**Generic**	**Exploits**	**Fuzzers**	**DoS**
STL	Precision	0.700	0.980	0.570	0.660	0.320
	Recall	0.980	0.970	0.680	0.040	0.430
	f1-score	0.820	0.980	0.620	0.070	0.370
H-MTL	Precision	0.993	0.980	0.918	0.959	0.114
	Recall	0.981	0.939	0.348	0.445	0.829
	f1-score	0.987	0.959	0.505	0.607	0.200
	False alarm	0.003	0.002	0.007	0.002	0.165
S-MTL	Precision	0.991	0.895	0.618	0.743	0.479
	Recall	0.996	0.450	0.684	0.708	0.403
	f1-score	0.994	0.598	0.649	0.725	0.438
	False alarm	0.0006	0.025	0.049	0.068	0.103
		Reconnaissance	**Analysis**	**Backdoor**	**Shellcode**	**Worms**
STL	Precision	0.770	0.000	0.000	1.000	0.620
	Recall	0.010	0.000	0.000	0.000	0.140
	f1-score	0.020	0.000	0.000	0.000	0.230
H-MTL	Precision	0.767	0.139	0.049	0.613	0.395
	Recall	0.478	0.407	0.315	0.192	0.769
	f1-score	0.589	0.207	0.084	0.292	0.522
	False alarm	0.0776	0.029	0.0619	0.008	0.001
S-MTL	Precision	0.995	0.249	0.156	0.005	0.050
	Recall	0.982	0.776	0.453	0.592	0.858
	f1-score	0.988	0.377	0.232	0.011	0.095
	False alarm	0.0010	0.048	0.089	0.081	0.103

и уровнем ложных тревог. H-MTL демонстрирует плохой уровень обнаружения – 19,2%, но относительно низкий уровень ложных тревог – 0,8%, в то время как S-MTL показывает гораздо более высокий уровень обнаружения – 59,2%, но также и гораздо более высокий уровень ложных тревог – 8,1%. Атаки Backdoor следуют аналогичной схеме. Тем не менее, S-MTL обеспечивает надежную работу с высокими показателями обнаружения и низким уровнем ложных тревог для классов разведки и анализа, что доказывает, что подход к мягкому совместному использованию параметров более эффективен для обнаружения редких атак.

При использовании настройки бинарной классификации показатели accuracy различных моделей следующие: для STL – 88,06%, для H-MTL – 88,28% и для S-MTL – 90,43%.

В экспериментах, проведенных с набором данных CICIDS2017, наблюдается, что, несмотря на крайний дисбаланс классов, задача классификации не так сложна, как с набором данных UNSW-NB15, и общая точность намного выше (таблица 5). Это объясняется тем, что в UNSW-NB15 встречаются редкие атаки с нулевым уровнем обнаружения для UNSW-NB15.

Однако эффективность по-прежнему варьируется. Модель STL обеспечивает высокие значения показателя precision во всех классах, особенно для нормального трафика, что указывает на то, что она имеет относительно более лучшую чувствительность при выявлении вредоносного поведения от нормального. С другой стороны, модель H-MTL также демонстрирует высокую точность для нормального трафика, DDoS-атак и атак грубой силы (Brute Force), но существует очевидный компромисс между способностью обнаружения и чувствительностью: precision – 10,9% и recall – 9,25%. Модель S-MTL не отличается чрезвычайно высокими значениями показателя precision, но, в целом, эти значения являются приемлемыми.

Хотя все модели демонстрируют приемлемую эффективность при нормальном трафике и атаках мажоритарных классов, таких как DoS, DDoS и Brute Force, обе модели MTL превосходят STL, особенно в классах Web Attack и Bot, а также в классе Infiltration. S-MTL не только достигает высоких показателей

Таблица 5

Оценки показателей классификации для набора данных CICIDS2017

		Normal	**DoS**	**DDoS**	**Brute Force**
STL	Precision	1.00	0.95	0.96	0.98
	Recall	0.99	0.99	0.99	0.99
	f1-score	0.99	0.98	0.98	0.98
H-MTL	Precision	0.999	0.975	0.986	0.986
	Recall	0.987	0.999	0.989	0.994
	f1-score	0.993	0.987	0.987	0.990
	False alarm	0.070	0.0001	0.0001	0.0006
S-MTL	Precision	0.994	0.995	0.740	0.980
	Recall	0.996	0.996	0.970	0.980
	f1-score	0.995	0.995	0.839	0.980
	False alarm	0.0005	0.0001	0.010	0.001
		Web Attack	**Bot**	**Infiltration**	**Heartbleed**
STL	Precision	1.000	0.85	0.83	0.42
	Recall	0.07	0.63	0.42	1.00
	f1-score	0.13	0.73	0.56	0.59
H-MTL	Precision	0.109	0.697	0.435	0.182
	Recall	0.825	0.801	0.633	1.000
	f1-score	0.193	0.745	0.516	0.308
	False alarm	0.0065	0.00025	0.00002	0.00002
S-MTL	Precision	0.802	0.733	0.357	0.182
	Recall	0.812	0.824	0.633	1.000
	f1-score	0.807	0.776	0.457	0.308
	False alarm	0.0002	0.0003	0.00003	0.00002

DOI: 10.21681/2311-3456-2024-2-10-21

обнаружения редких атак, но также имеет самые высокие оценки F1 для веб-атак и ботов, что указывает на сбалансированную эффективность обнаружения (по значениям precision и recall) для этих редких атак. Аналогичным образом, H-MTL демонстрирует высокие оценки F1 для атак ботов и Infiltration, но для атаки Infiltration существует некоторый компромисс между precision и recall. Однако для Heartbleed модели MTL получили низкие оценки. Имея в тестовом наборе всего четыре экземпляра (см. таблицу 4), все три модели могут успешно обнаружить эту атаку; однако обе модели MTL демонстрируют более низкие значения precision, чем модель STL.

5. Заключение

В статье представлена реализация классификатора сетевого трафика для обнаружения атак в сети Интернета вещей на основе многозадачного обучения с двумя различными механизмами совместного использования параметров. Предложенный подход основан на ряде новых решений и обладает несколькими важными преимуществами: (1) предложен метод оптимизации потерь для конкретной задачи за счет включения оптимальной функции потерь и функции оптимизации веса для конкретной задачи; (2) предложен метод гибридного сэмплирования, использующий как случайную субдискретизацию, так и передискретизацию на основе GAN, которая сочетает в себе традиционные методы и генеративную модель на основе глубокого обучения; (3) проведен анализ эффективности обнаружения, и осуществлена проверка возможности обобщения предложенных решений на двух разных наборах данных.

Результаты экспериментов показали, что многозадачное обучение превосходит однозадачное, особенно при обнаружении редких классов атак. Несмотря на то, что баланс между показателями обнаружения и ложными тревогами необходимо улучшить (что является задачей последующих исследований), предложенный подход представляет собой практическую основу, предназначенную для реализации робастных механизмов обнаружения атак в сетях Интернета вещей.

Рецензент: *Лаута Олег Сергеевич, доктор технических наук, профессор кафедры комплексного обеспечения информационной безопасности Государственного университета морского и речного флота имени адмирала С. О. Макарова, Санкт-Петербург, Россия. E mail: laos-82@yandex.ru*

Литература

1. *Jurcut A. D., Ranaweera P., Xu L. Introduction to IoT security // IoT security: advances in authentication. 2020. P. 27–64.*
2. *Hussain F., Hussain R., Hassan S. A., Hossain E. Machine learning in IoT security: Current solutions and future challenges // IEEE Communications Surveys & Tutorials. 2020 № 22(3). P. 1686–721.*
3. *Kotenko I., Saenko I., Privalov A., Lauta O. Ensuring SDN Resilience under the Influence of Cyber Attacks: Combining Methods of Topological Transformation of Stochastic Networks, Markov Processes, and Neural Networks // Big Data and Cognitive Computing, 2023. 7(2): 66. P.1–39.*
4. *Котенко И. В., Левшун Д. А. Методы интеллектуального анализа системных событий для обнаружения многошаговых кибератак: использование методов машинного обучения // Искусственный интеллект и принятие решений, 2023, № 3. С.3–16.*
5. *Котенко И. В., Саенко И. Б., Аль-Барри М. Х. Выявление аномального поведения пользователей центров обработки данных вузов // Правовая информатика, 2023. № 1. С.62–71. DOI: 10.21681/1994-1404-2023-1-62-71.*
6. *Zhang Y., Yang Q. A survey on multi-task learning // IEEE transactions on knowledge and data engineering. 2022. № 34. P. 5586–5609.*
7. *Dong H., Kotenko I. Intrusion Detection with Uncertainty based Loss Optimized Multi-Task Learning // In Proceedings of the International Conference on Information Processes and Systems Development and Quality Assurance. 2023. P. 69–73.*
8. *Dong H., Kotenko I. An Autoencoder-based Multi-task Learning for Intrusion Detection in IoT Networks // 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2023. 4 p.*
9. *Lan J., Liu X., Li B., Sun J., Li B., Zhao J. MEMBER: A multi-task learning model with hybrid deep features for network intrusion detection // Computers & Security. 2022. № 123. P. 102919.*
10. *Liu Q., Wang D., Jia Y., Luo S., Wang C. A multi-task based deep learning approach for intrusion detection // Knowledge-Based Systems. 2022. № 238. P. 107852.*
11. *Mustafa M., Buttar A. M., Sajja G. S., Gour S., Naved M., William P. Multitask Learning for Security and Privacy in Iov (Internet of Vehicles) // Autonomous Vehicles Volume 1: Using Machine Intelligence. 2022. P. 217–233.*
12. *Hamdan S., Almajali S., Ayyash M., Salameh H. B., Jararweh Y. An intelligent edge-enabled distributed multi-task learning architecture for large-scale IoT-based cyber–physical systems // Simulation Modelling Practice and Theory. 2023. № 122. P. 102685.*
13. *Andresini G., Appice A., De Rose L., Malerba D. GAN augmentation to deal with imbalance in imaging-based intrusion detection // Future Generation Computer Systems. 2021. № 123. P. 108–27.*
14. *Mushtaq E., Zameer A., Umer M., Abbasi AA. A two-stage intrusion detection system with auto-encoder and LSTMs // Applied Soft Computing. 2022. № 1(121). P. 108768.*
15. *Yue C., Wang L., Wang D., Duo R., Nie X. An ensemble intrusion detection method for train ethernet consist network based on CNN and RNN // IEEE Access. 2021. № 15(9). P. 59527.*

16. *Siddiqui A. J., Boukerche A. Adaptive ensembles of autoencoders for unsupervised IoT network intrusion detection // Computing. 2021. № 103(6). P. 1209.*
17. *Liu J., Zhang W., Ma T., Tang Z., Xie Y, Gui W, Niyoyita JP. Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection // Expert Systems with Applications. 2020. № 15(158). P. 113578.*
18. *Omuya E. O., Okeyo G. O., Kimwele M. W. Feature Selection for Classification using Principal Component Analysis and Information Gain // Expert Systems with Applications. Vol.174. July 2021. 114765.*
19. *Zhang Y., Yang Q. A survey on multi-task learning // IEEE Transactions on Knowledge and Data Engineering. Vol.34, Iss.12, 2022. P. 5586–5609.*
20. *Kasongo S. M., Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset // Journal of Big Data, Vol.7, 2020. 105.*
21. *Neto E E.C.P., Dadkhah S., Ferreira R., Zohourian A., Lu R., Ghorbani A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment // Sensors. 2023. № 23(13). P. 5941.*
22. *Barakat A., Bianchi P. Convergence and dynamical behavior of the ADAM algorithm for nonconvex stochastic optimization // SIAM Journal on Optimization. Vol.31. Iss.1. 2021.*