БЕЗОПАСНОСТЬ ЦИФРОВЫХ ТЕХНОЛОГИЙ. – 2024. – № 4 (115). – 37–65

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.8 DOI: 10.17212/2782-2230-2024-4-37-65

ПРИМЕНЕНИЕ НЕЙРОННОЙ СЕТИ MULTILAYER PERCEPTRON (MLP) ДЛЯ ОБНАРУЖЕНИЯ И КЛАССИФИКАЦИИ КИБЕРУГРОЗ В СЕТЕВОМ ТРАФИКЕ*

А.Г. ПОДСЕВАЛОВ[1], М.А. КИСЕЛЕВ[2], А.В. ИВАНОВ[3]

[1] *630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры защиты информации. E-mail podsevalov.2019@stud.nstu.ru*
[2] *630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры защиты информации. E-mail: m.kiselev.2019@stud.nstu.ru*
[3] *630073, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент, заведующий кафедрой защиты информации. E-mail: andrej.ivanov@corp.nstu.ru*

В настоящей статье рассматривается применение многослойного перцептрона (MLP) для классификации сетевого трафика с целью обнаружения киберугроз. Модель была обучена на датасете NSL-KDD, который является стандартом для задач выявления атак и широко используется в исследованиях. В ходе экспериментов была проведена предварительная обработка данных, включающая кодирование категориальных признаков и балансировку классов методом SMOTE для устранения дисбаланса между нормальным и вредоносным трафиком. Результаты показали высокую точность классификации – 96.64 % – даже в условиях формирования шума и 10-кратной кросс-валидации, что подтверждает надежность предложенного подхода. В статье предложены показатели эффективности, такие как точность, полнота и F1-мера, которые могут служить основой для дальнейших исследований и оптимизации моделей машинного обучения для повышения безопасности сетей.

Ключевые слова: информационная безопасность, киберугрозы, машинное обучение, MLP, NSL-KDD, ROC-AUC, SMOTE, кросс-валидация, One-Hot Encoding

* Статья получена 20 ноября 2024 г.

ВВЕДЕНИЕ

С увеличением объема сетевого трафика и усложнением кибератак традиционные методы обнаружения сталкиваются с ограничениями, такими как низкая эффективность при работе с новыми и модифицированными атаками, а также неспособность обрабатывать большие объемы данных в реальном времени. Киберугрозы эволюционируют, становясь более изощренными и целенаправленными, что значительно усложняет их обнаружение с использованием традиционных методов [1]. Средства защиты информации (СЗИ) на основе сигнатурного анализа и системы обнаружения на основе правил имеют ряд ограничений. Сигнатурные методы хорошо справляются с известными угрозами, но они беспомощны перед новыми или модифицированными атаками [2]. Методы, основанные на правилах, требуют постоянного обновления и часто неэффективны против сложных многоступенчатых атак или тех, которые используют обфускацию [3].

Эти проблемы обостряются в условиях увеличения объема и сложности сетевого трафика. Возникает необходимость в применении интеллектуальных и адаптивных методов для автоматического анализа сетевого трафика и своевременного выявления аномалий.

Методы машинного обучения, особенно глубокого обучения, демонстрируют большой потенциал в решении этих задач. Многослойный персептрон (MLP), являющийся одной из базовых архитектур глубокого обучения, способен моделировать сложные нелинейные зависимости в сетевом трафике, выявлять скрытые аномалии и автоматизировать процесс классификации угроз. Это делает MLP перспективным инструментом для обнаружения ранее неизвестных атак и повышения точности систем защиты. В отличие от других архитектур, таких как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), MLP проще в настройке и не требует сложных данных с локальной структурой (для CNN) или временной зависимостью (для RNN), что делает его более гибким и подходящим для разнообразных данных сетевого трафика [4].

Цель настоящей работы заключается в исследовании возможностей применения многослойного персептрона (MLP) для выявления киберугроз путем классификации сетевого трафика. Для достижения этой цели в исследовании рассматриваются несколько ключевых аспектов. В первую очередь проводится обзор существующих методов, применяемых для обнаружения киберугроз, что позволяет лучше понять их ограничения и преимущества. Далее внимание уделяется описанию методов предобработки данных, которые необходимы

для корректной работы MLP, включая нормализацию и кодирование данных. Особое внимание уделяется выбору информативных признаков, которые играют решающую роль в процессе обучения модели. На следующем этапе осуществляется практическая реализация модели на основе открытых датасетов, что позволяет проверить ее работоспособность в условиях реальных данных. Для оценки эффективности модели проводится тщательная оценка ее производительности, включая такие метрики, как точность, полнота и $F1$-меры. Наконец, результаты анализируются с целью выявления существующих ограничений модели и предложений по ее улучшению в будущем.

1. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ В ОБНАРУЖЕНИИ И КЛАССИФИКАЦИИ КИБЕРУГРОЗ

Современные системы обнаружения киберугроз зачастую используют традиционные методы, такие как сигнатурный подход и анализ, основанный на правилах [5]. Эти подходы показали себя как эффективные для выявления известных атак и отклонений в сетевом трафике. Однако с ростом сложности и изменчивости кибератак эти методы сталкиваются с рядом ограничений, которые затрудняют их способность к обнаружению неизвестных аномалий.

Кроме того, для успешного реагирования на кибератаки важно не только идентифицировать угрозу, но и корректно классифицировать ее тип, что позволит предпринять меры по защите. Традиционные системы классификации угроз часто ограничены статическими правилами или базами данных известных угроз [6]. Это может приводить к ошибкам при классификации сложных или многоступенчатых атак, использующих обфускацию или комбинированные методы. В таких условиях интеллектуальные системы, использующие методы машинного обучения, способны обеспечить не только обнаружение угроз, но и их точную классификацию.

Многослойный персептрон (MLP) благодаря своей способности моделировать сложные нелинейные зависимости предлагает адаптивный подход к задаче классификации киберугроз [16]. Обучаясь на данных, модель автоматически выделяет паттерны для разных типов атак, что позволяет значительно повысить точность как в обнаружении, так и в классификации угроз. Это делает MLP более эффективным по сравнению с сигнатурным подходом, который ограничен в распознавании новых и неизвестных атак [7].

2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ MLP И ЕГО ПРЕИМУЩЕСТВА ДЛЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ

2.1. ОСНОВЫ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА

Многослойный персептрон (MLP) является базовой архитектурой нейронной сети, способной решать задачи классификации и регрессии благодаря своей способности моделировать сложные нелинейные зависимости. В кибербезопасности MLP может применяться для классификации сетевого трафика, обнаружения аномалий и киберугроз, включая новые и сложные виды атак[8].

Архитектура MLP включает три основных компонента (рис. 1) [9].

1. **Входной слой (Input Layer).** Принимает набор признаков (вектор данных), представляющий сетевой трафик:

- **Input 1** – объем трафика (например, количество байтов, переданных за определенное время);
- **Input 2** – тип протокола (например, TCP, UDP, ICMP);
- **Input 3** – время между отправкой пакетов (интервал времени между последовательными пакетами).

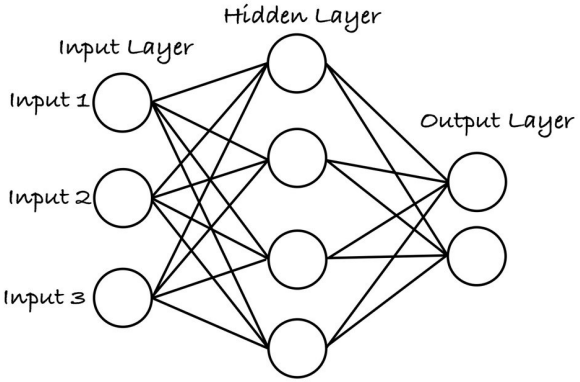

Рис. 1. Архитектура MLP

2. Каждый нейрон входного слоя соответствует одному признаку (например, объем трафика, протокол соединения и т. д.).

3. **Скрытые слои (Hidden Layer).** Содержат один или несколько слоев нейронов, которые обрабатывают информацию, поступающую из входного

слоя. Именно в этих слоях происходит моделирование сложных нелинейных зависимостей между признаками. Каждый нейрон скрытого слоя вычисляет взвешенную сумму своих входных значений и применяет к ним функцию активации.

4. **Выходной слой (Output Layer).** Генерирует предсказание (например, классификацию сетевого трафика как нормальный или вредоносный). В зависимости от задачи выходной слой может использовать одну или несколько функций активации.

Функции активации являются важнейшей частью проектирования нейронной сети. Выбор функции активации в скрытом слое определяет, насколько хорошо модель усваивает обучающий набор данных, а в выходном слое определяет тип прогнозов, которые может делать модель. Ниже представлены основные функции активации [10]:

- **ReLU (Rectified Linear Unit)** – одна из самых популярных функций активации в скрытых слоях. Среди достоинств более эффективная оптимизация и уменьшение проблемы исчезающего градиента. Математически функция задана следующим образом:

$$f(x) = \max(0, x). \tag{1}$$

- **Sigmoid**. Сигмовидная функция нелинейна и ограничивает значение нейрона в небольшом диапазоне (0; 1). Часто используется в выходном слое для задач бинарной классификации. Математическая функция

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

где e – экспоненциальная функция.

- **Softmax**. Функция активации softmax преобразует необработанные выходные данные нейронной сети в вектор вероятностей (распределение вероятностей по входным классам). Математическая функция

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}, \tag{3}$$

где z_i – входное значение для класса i; K – общее количество классов; e – экспоненциальная функция.

Softmax используется в выходном слое многослойного персептрона для задач многоклассовой классификации, в которых модель должна предсказать вероятность принадлежности объекта к каждому из K классов.

2.2. АЛГОРИТМ ОБУЧЕНИЯ MLP

Основной алгоритм, используемый для обучения MLP, – это алгоритм обратного распространения ошибки **(Backpropagation)**, который вычисляет градиенты функции потерь относительно параметров модели и обновляет параметры итеративно в сочетании с методом градиентного спуска для обновления весов сети [13, 14].

1. В методе прямого распространения **(Forward Propagation)** каждый нейрон вычисляет взвешенную сумму своих входов, применяет функцию активации и передает результат следующему слою. Также вычисляется функция потерь, которая измеряет разницу между предсказанными выходными данными и фактическими целевыми значениями.

2. Алгоритм обратного распространения ошибки **(Backpropagation)** вычисляет градиенты функции потерь относительно каждого веса и смещения в сети.

3. Далее веса и смещения обновляются на основе вычисленных градиентов с помощью метода градиентного спуска **(Gradient Descent)**. Это подразумевает выполнение шага в направлении, которое уменьшает функцию потерь. Формула обновления весов выглядит следующим образом:

$$W_{ij} = W_{ij} - \eta \frac{\partial L}{\partial W_{ij}}, \tag{4}$$

где W_{ij} – вес между нейронами i и j; η – скорость обучения; $\frac{\partial L}{\partial W_{ij}}$ – градиент функции потерь L по отношению к весам.

Чтобы избежать переобучения (модель слишком точно подстраивается под обучающие данные и теряет способность обобщать новые данные), применяются методы регуляризации ($L1$ и $L2$ – от англ. *Lasso regression*) [24].

- Регуляризация $L1$ добавляет штраф за сумму абсолютных значений весов, что помогает обнулить несущественные веса, тем самым уменьшая сложность модели.

- Регуляризация *L2* добавляет штраф за сумму квадратов весов, что позволяет сглаживать веса и предотвращает слишком сильные изменения весов нейронов.

Блок-схема, изображающая алгоритм обучения MLP, представлена на рис. 2.

Рис. 2. Алгоритм обучения MLP

2.3. ПРЕИМУЩЕСТВА MLP ДЛЯ ОБНАРУЖЕНИЯ КИБЕРУГРОЗ

При анализе сетевого трафика MLP преобразует данные о трафике в вектор признаков, отражающих различные характеристики сетевых пакетов, такие как время передачи, размер, тип протокола и другие важные метрики [11]. Эти признаки затем проходят через несколько слоев нейронной сети, в которой модель обучается выявлять сложные нелинейные зависимости и паттерны, присущие различным типам атак.

Благодаря способности MLP распознавать скрытые аномалии и отличительные особенности вредоносного трафика модель эффективно классифицирует различные виды киберугроз. В распознавание аномалий входит обнаружение вредоносных атак, таких как DDoS-атаки, сканирование портов и других аномалий в сетевом трафике. В связи с этим применение MLP в области кибербезопасности может позволить не только обнаруживать известные типы атак, но и выявлять новые и ранее неизвестные угрозы путем повторного обучения модели, тем самым повышая общую эффективность системы защиты [12]. Ряд ключевых преимуществ [15] MLP:

1) выявление сложных нелинейных зависимостей: MLP способен выявлять нелинейные зависимости между признаками сетевого трафика [16], это позволяет обнаруживать аномалии и скрытые паттерны, которые могут указывать на подозрительную активность в сетевом трафике;

2) распознавание новых угроз: в отличие от сигнатурных методов, MLP обучается на реальных данных, что позволяет распознавать новые или измененные угрозы, анализируя уникальные аномалии в трафике [17];

3) гибкость и адаптивность: MLP можно переобучить на новых данных, что позволяет модели адаптироваться к постоянно изменяющимся атакам [18];

4) обработка больших объемов данных: MLP достаточно хорошо справляется с обработкой больших объемов данных благодаря возможности параллельных вычислений и оптимизированным алгоритмам обучения [19];

5) применимость к различным видам атак, включая DDoS-атаки, ботнеты, фишинг и др. Благодаря обучению на разных типах данных модель способна адаптироваться к анализу различных типов атак [20].

3. МЕТОДЫ ПРЕДОБРАБОТКИ ДАННЫХ ДЛЯ MLP

Для успешного обучения и эффективного функционирования многослойного персептрона в задаче классификации сетевого трафика на нормальный и вредоносный необходима предварительная предобработка данных.

Сетевой трафик, представляющий собой большой массив разнообразной информации, включает как числовые, так и категориальные признаки, и требует адаптации перед подачей в нейронную сеть.

Основные шаги предобработки данных для MLP [21].

1. **Очистка данных** – удаление дубликатов, пропущенных или некорректных значений.

2. **Нормализация и масштабирование** – приведение числовых признаков к единому диапазону (например, с помощью нормализации минимакса или Z-score) для ускорения и стабилизации обучения модели.

3. **Кодирование категориальных признаков** – преобразование категориальных данных в числовой формат, например, с помощью One-Hot Encoding.

4. **Обработка несбалансированных данных** – применение способов балансировки классов (например, oversampling или undersampling) для устранения дисбаланса между нормальным и вредоносным трафиком.

5. **Выбор и создание информативных признаков** – отбор наиболее значимых признаков и создание новых характеристик для улучшения качества модели и повышения ее способности различать нормальные и аномальные паттерны.

После успешной предобработки данных MLP может эффективно обучаться и использовать эти данные для классификации сетевого трафика.

4. КРИТЕРИИ ОЦЕНКИ РАБОТЫ МОДЕЛИ MLP В КОНТЕКСТЕ АНАЛИЗА СЕТЕВОГО ТРАФИКА

4.1. ТОЧНОСТЬ (ACCURACY)

Точность (Accuracy) показывает долю корректно классифицированных данных среди общего числа примеров. Однако в случае несбалансированных данных, таких как сетевой трафик, этот показатель может оказаться неинформативным, поскольку модель может демонстрировать высокую точность, просто предсказывая все примеры как нормальные:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

где TP – True Positives (правильные предсказания атак); TN – True Negatives (правильные предсказания нормального трафика); FP – False Positives

(ложноположительные срабатывания); *FN* – False Negatives (ложноотрицательные срабатывания).

4.2. ПОЛНОТА (RECALL)

Полнота (Recall) модели показывает, насколько она способна обнаруживать все атаки в общем объеме вредоносного трафика. Стоит отметить, что модель с высокой полнотой может давать больше ложных сигналов о наличии атак:

$$Recall = \frac{TP}{TP + FN}. \tag{6}$$

4.3. ТОЧНОСТЬ ПРЕДСКАЗАНИЙ (PRECISION)

Точность предсказаний (Precision) показывает, насколько предсказанные моделью атаки действительно являются вредоносными. Высокая точность предсказаний важна для снижения числа ложных срабатываний (когда модель ошибочно классифицирует нормальный трафик как атаку):

$$Precision = \frac{TP}{TP + FP}. \tag{7}$$

4.4. *F*1-МЕРА

Это гармоническое среднее между полнотой и точностью предсказаний. *F*1-мера будет необходима в условиях несбалансированных данных, когда нужно будет учитывать баланс между полнотой и точностью:

$$F1 = 2\,\frac{Precision \times Recall}{Precision + Recall}. \tag{8}$$

4.5. ROC-КРИВАЯ И ПЛОЩАДЬ ПОД КРИВОЙ (AUC)

ROC-кривая (Receiver Operating Characteristic) представляет собой отображение зависимости между долей верных срабатываний (True Positive Rate) и долей ложных (False Positive Rate) при различных порогах классификации.

Площадь под кривой (AUC) является обобщающей метрикой, характеризующей качество модели. Чем ближе значение AUC к единице, тем лучше модель способна различать нормальный и вредоносный трафик.

ROC-кривая формируется на основе точек, соответствующих соотношениям истинно положительных срабатываний (True Positive Rate) и ложноположительных срабатываний (False Positive Rate) на различных порогах:

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN}. \tag{9}$$

ROC-кривая отображает способность модели различать классы. Чем выше кривая, тем лучше модель справляется с задачей классификации.

Площадь под ROC-кривой (AUC) вычисляется как интеграл множества точек под ROC-кривой. Значение AUC, близкое к единице, свидетельствует о том, что модель хорошо различает классы:

$$AUC = \int_0^1 TPRd\,(FPR). \tag{10}$$

AUC считается одной из наиболее универсальных метрик для оценки моделей, так как она не зависит от дисбаланса классов.

Вышеописанные критерии являются ключевыми для оценки производительности модели MLP и в дальнейшем будут использоваться в расчете оценки точности данной модели.

5. ПЕРЕХОД К ПРОЦЕССУ ОБУЧЕНИЯ И ОПТИМИЗАЦИИ МОДЕЛИ

После оценки модели MLP на основе вышеописанных критериев становится понятно, что оценка качества модели – это лишь начало работы. Чтобы достичь высокой точности классификации и обнаружения атак по этим критериям, необходимо тщательно разработать процесс обучения и оптимизации модели.

Одной из главных задач при обучении MLP является минимизация функции потерь, что напрямую влияет на качество предсказаний модели. Для этого применяются методы «градиентный спуск» и «регуляризация», которые не только улучшают точность модели, но и предотвращают такие проблемы, как переобучение.

Рассмотрим ключевые аспекты процесса обучения и оптимизации модели MLP, которые позволяют достичь максимальных результатов по метрикам качества классификации.

5.1. ФУНКЦИЯ ПОТЕРЬ И ЕЕ МИНИМИЗАЦИЯ

Для многоклассовой классификации часто используется кросс-энтропия как функция потерь [20, 21]. Она измеряет разницу между истинными метками классов и предсказанными вероятностями:

$$L = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log\left(\hat{y}_{ik}\right), \tag{11}$$

где N – количество примеров; K – количество классов; y_{ik} – истинная метка класса k для примера i; $\hat{y}_{ik}$ – предсказанная вероятность класса k для примера i.

Выбор функции потерь, основанной на кросс-энтропии, играет ключевую роль в процессе обучения многослойного перцептрона (MLP) для задач классификации. Минимизация этой функции позволяет модели предсказывать вероятности классов, что приводит к улучшению таких метрик качества, как $F1$-мера и AUC.

Применение градиентного спуска в сочетании с регуляризацией способствует повышению точности модели и предотвращает переобучение, обеспечивая баланс между сложностью модели и ее способностью обобщать новые данные.

5.2. ГРАДИЕНТНЫЙ СПУСК И ОБНОВЛЕНИЕ ВЕСОВ

Оптимизация сети MLP основана на градиентном спуске для минимизации функции потерь [20, 21]. Алгоритм вычисляет градиенты функции потерь по каждому весу и обновляет веса в направлении уменьшения ошибки модели согласно формуле (4).

Основные шаги алгоритма (рис. 3):

1) **прямое распространение (Forward pass)**. Входные данные проходят через сеть, вычисляется предсказание $\hat{y}$;

2) **вычисление функции потерь**. Вычисляется функция потерь L между истинными метками y и предсказанными значениями $\hat{y}$;

3) **обратное распространение (Backward pass)**. Вычисляются градиенты функции потерь по весам, начиная с выходного слоя в обратную сторону;

4) **обновление весов**. Веса обновляются с использованием вычисленных градиентов по формуле градиентного спуска.

Этот процесс представляет собой итеративное вычисление градиентов и последующее обновление весов с целью минимизации ошибки. Скорость обу-

чения η определяет степень изменения весов на каждой итерации: малые значения способствуют более плавному, но медленному обучению, в то время как большие значения могут ускорить сходимость, но привести к нестабильности.

Цикл вычисления градиентов и обновления весов повторяется для всего обучающего набора до тех пор, пока функция потерь не достигнет оптимального значения на обучающих данных.

5.3. РЕГУЛЯРИЗАЦИЯ ДЛЯ ПРЕДОТВРАЩЕНИЯ ПЕРЕОБУЧЕНИЯ

Чтобы модель не переобучалась на тренировочных данных, используют методы регуляризации. Один из наиболее популярных методов – *L*2-регуляризация (или Ridge-регуляризация):

$$L = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log\left(\hat{y}_{ik}\right) + \lambda \sum_{j=1}^{M} W_j^2 , \tag{12}$$

где λ – коэффициент регуляризации, который контролирует влияние штрафа; M – количество классов; W_j – веса модели.

Итоговая схема формирования алгоритма обучения многослойного персептрона (MLP) с использованием градиентного спуска и обратного распространения ошибки представлена на рис. 3.

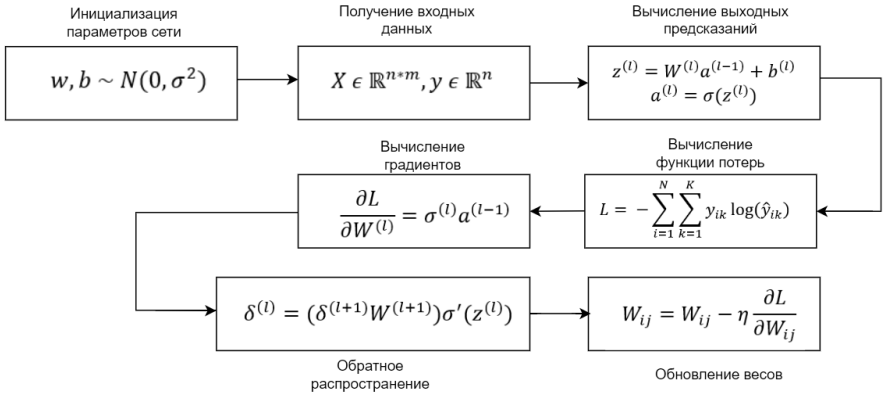

Рис. 3. Алгоритм обучения многослойного персептрона (MLP) с использованием градиентного спуска и обратного распространения ошибки

Таким образом, были рассмотрены основные аспекты процесса обучения и оптимизации модели: минимизация функции потерь (кросс-энтропия) с помощью градиентного спуска и алгоритма обратного распространения ошибки, а также применение регуляризации ($L2$-регуляризация) для предотвращения переобучения. Эти методы позволяют модели более точно предсказывать классы и эффективно работать с новыми данными.

6. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ОЦЕНКА ЭФФЕКТИВНОСТИ MLP НА ДАТАСЕТЕ NSL-KDD

Для реализации практической части использовался датасет NSL-KDD, представляющий собой улучшенную версию популярного набора данных KDD Cup 99, который применяется для обнаружения сетевых атак [22]. Целью эксперимента было создание и обучение модели многослойного перцептрона (MLP) для классификации сетевого трафика на нормальные соединения и соединения, содержащие кибератаки.

6.1. ПРЕОБРАЗОВАНИЕ ДАННЫХ

Для успешного применения модели глубокого обучения необходимо было предварительно обработать данные.

1. One-Hot Encoding применялся для категориальных признаков, таких как тип протокола, служба, флаг соединения [23].

2. Нормализация числовых признаков с помощью стандартизации для улучшения сходимости модели [24].

3. SMOTE (Synthetic Minority Over-sampling Technique) был использован для балансировки классов в обучающей выборке, так как атакующие соединения встречаются реже, чем нормальные [25].

6.2. ОБУЧЕНИЕ МОДЕЛИ MLP

Модель многослойного перцептрона (MLP) с архитектурой из двух скрытых слоев (50 и 25 нейронов соответственно) была обучена на обработанном датасете. Выбор 50 и 25 нейронов – это стандартная конфигурация для моделей средней сложности. Она достаточно мощная, чтобы захватить важные паттерны в данных, но при этом не слишком сложная, что может привести к переобучению [26, 27]. Если увеличить количество нейронов, модель может быть слишком сложной для данных, что может ухудшить ее способность

обобщать новые данные. Обучение выполнялось с использованием стохастического градиентного спуска с максимальным количеством итераций 1000:

```
mlp = MLPClassifier(hidden_layer_sizes=(50, 25), max_iter=1000, alpha=0.001, random_state=42)
```

Для разбиения данных использовалось такое соотношение: 80 % для обучения и 20 % для тестирования [26]. Пример датасета NSL-KDD, используемого для обнаружения сетевых атак, где каждая строка представляет собой сетевое соединение и включает в себя набор признаков, разделенных запятыми, см. в табл. 1.

Таблица 1

Датасет NSL-KDD

Тип протокола	Тип службы	3	4	5	6	7	8	9	10	11	Метка класса	Время соединения
tcp	ftp_data	SF	491	0	0	0,17	0,00	0,00	0,05	0,00	normal	20
udp	other	SF	146	0	0	0,88	0,00	0,00	0,00	0,00	normal	15
tcp	private	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	19
tcp	http	SF	232	8153	0	0,03	0,03	0,01	0,00	0,01	normal	21
tcp	http	SF	199	420	0	0,00	0,00	0,00	0,00	0,00	normal	21
tcp	private	REJ	0	0	0	0,00	0,00	0,00	1,00	1,00	neptune	21
tcp	private	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	21
tcp	private	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	21
tcp	remote_job	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	21
tcp	private	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	21
tcp	private	REJ	0	0	0	0,00	0,00	0,00	1,00	1,00	neptune	21
tcp	private	S0	0	0	0	0,00	1,00	1,00	0,00	0,00	neptune	21
tcp	http	SF	287	2251	0	0,12	0,00	0,00	0,00	0,00	normal	21

Первые несколько столбцов содержат информацию о сетевых характеристиках соединения:

- **тип протокола** (например, TCP, UDP);
- **тип службы** (например, ftp_data, HTTP, private);
- другие количественные и категориальные признаки, характеризующие сетевую активность, такие как размер пакетов, количество ошибок и т. д.

Цифрами от 3 до 11 в табл. 1 обозначены: флаг соединения TCP(3); число переданных байтов от источника к назначению и наоборот (4,5); число ошибочных фрагментов (6); число сегментов, пересылаемых по TCP (7); процент

потерь от источника к назначению и наоборот (8, 9); процент передачи с ошибками (10); процент повторных передач (11).

Последние два столбца содержат:

- метку класса (например, normal, neptune), которая указывает, является соединение нормальным или атакой;
- время соединения (например, 20, 21), которое также может использоваться как признак.

Выбранная архитектура позволит эффективно захватить важные паттерны в данных без переобучения. Обучение с использованием стохастического градиентного спуска и максимальным количеством итераций 1000 должно обеспечить стабильную сходимость модели.

6.2. ОБУЧЕНИЕ МОДЕЛИ MLP

В ходе исследования было обучено две модели многослойного перцептрона (MLP) на датасете NSL-KDD. Первая модель была обучена без каких-либо усложнений в данных, а вторая модель включала добавление шума в тренировочные и тестовые данные, а также использование метода кросс-валидации для улучшения общей оценки качества модели. Добавление шума в обучающие данные (с помощью переменной noise_factor) (рис. 5) использовалось для улучшения способности модели к обобщению и для повышения ее устойчивости к нестабильным данным, что часто встречается в реальных условиях работы с сетевым трафиком. Шум генерируется с использованием нормального распределения с математическим ожиданием $\mu = 0$ и стандартным отклонением $\sigma = 1{,}0$ [28]. Фактор шума (noise_factor) умножается на значения, полученные из нормального распределения, и добавляется к каждому элементу обучающих и тестовых данных (X_train и X_test) (рис. 5).

Для начала была обучена стандартная модель MLP без дополнительных модификаций. Эта модель была построена с двумя скрытыми слоями (50 и 25 нейронов) и с применением $L2$-регуляризации (с коэффициентом регуляризации 0,001). Данные для обучения и тестирования были разделены в соотношении 4 : 1. Балансировка классов была выполнена с помощью метода SMOTE для корректного представления атакующего и нормального трафика.

Результаты этой модели оказались практически идеальными. Точность классификации составила 99,72 %, а значение метрики ROC-AUC – 1,00, что свидетельствует о высоком качестве разделения нормального и вредоносного трафика. В отчете по классификации все метрики (Precision, Recall и $F1$-score) также приняли значение «1,00» для обоих классов. Это говорит о том, что модель не делала ошибок при предсказании нормальных соединений и атак.

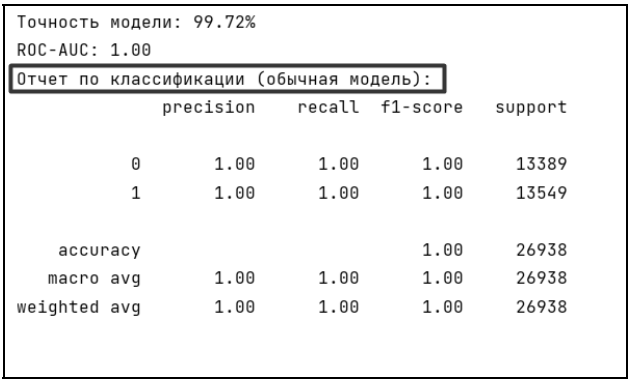

Рис. 4. Результат обучения модели многослойного перцептрона (MLP) на датасете NSL-KDD без шума

Таблица 2

Отчет по классификации метрик

Метрика	Класс 0 (нормальные)	Класс 1 (атаки)
Точность (Precision)	1,00	1,00
Полнота (Recall)	1,00	1,00
*F*1-мера	1,00	1,00
Число примеров (Support)	13,389	13,549

Отчет по классификации, включающий метрики Precision, Recall и *F*1-меру, был получен для двух классов:

1) класс 0 (нормальные соединения);

2) класс 1 (атаки).

Precision (точность предсказаний) показывает, какой процент соединений, предсказанных как «атака», действительно является атаками. В обоих классах (нормальные соединения и атаки) точность составила 1,00. Это означает, что модель предсказывала метки с высокой уверенностью.

Recall (полнота) определяет, какой процент реальных атак был корректно обнаружен. Полнота также составила 1,00. Это говорит о том, что модель не пропускала атаки и корректно их классифицировала.

***F*1-мера** – гармоническое среднее между Precision и Recall. Поскольку обе метрики (точность и полнота) идеальны, *F*1-мера также равна 1,00 для обоих классов.

Результаты показывают, что по всем трем метрикам точность составила 1,00. Такие идеальные показатели могут указывать на два возможных фактора: либо обучающие данные слишком просты, либо модель подверглась переобучению (Overfitting). Это означает, что модель, вероятно, «запомнила» данные, на которых обучалась, и поэтому демонстрирует высокие результаты, но может быть недостаточно устойчивой к новым данным.

Для проверки достоверности этих результатов будет использовано добавление шума в данные и метод кросс-валидации [26, 27]. Это поможет усложнить задачу, изменив структуру данных, и позволит оценить, как модель справляется с модифицированными данным.

В этой модели в обучающие и тестовые данные был добавлен небольшой шум (с коэффициентом шума 0,7), который имитирует реальные условия работы с зашумленными данными. Помимо этого, модель проходила через процесс кросс-валидации (для теста была 10-кратная кросс-валидация), что позволило избежать зависимости от конкретной обучающей выборки и проверить модель на нескольких разбиениях данных.

```
# Добавление шума в обучающие данные для усложнения задачи модели
noise_factor = 0.7
X_train_noisy = X_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=X_train.shape)
X_test_noisy = X_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=X_test.shape)

# Обучение модели на данных с шумом
mlp_noisy = MLPClassifier(hidden_layer_sizes=(50, 25), max_iter=1000, alpha=0.001, random_state=42)

# 5-кратная кросс-валидация для модели с шумом
cross_val_scores_noisy = cross_val_score(mlp_noisy, X_train_noisy, y_train, cv=10)
print(f"Средняя точность по кросс-валидации (с шумом): {np.mean(cross_val_scores_noisy) * 100:.2f}%")

# Обучение модели на данных с шумом
mlp_noisy.fit(X_train_noisy, y_train)

# Предсказание на зашумленных данных
y_pred_noisy = mlp_noisy.predict(X_test_noisy)
```

Рис. 5. Формирование кросс-валидации с шумом

```
Средняя точность по кросс-валидации (обычная модель): 99.71%
Средняя точность по кросс-валидации (с шумом): 97.00%
Точность модели с шумом: 96.94%
ROC-AUC (с шумом): 0.99
Отчет по классификации (с шумом):
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     13389
           1       0.97      0.97      0.97     13549

    accuracy                           0.97     26938
   macro avg       0.97      0.97      0.97     26938
weighted avg       0.97      0.97      0.97     26938
```

Рис. 6. Результат обучения модели многослойного перцептрона (MLP) на датасете NSL-KDD с шумом и кросс-валидацией

Полученные результаты точности модели слегка отличаются в худшую сторону, но это было ожидаемо, так как добавление шума и изменение кросс-валидации усложнило задачу для данной модели данных. Результаты метрик Precision, Recall и *F*1-score уменьшились для обоих классов, что говорит о небольшой неустойчивости модели MLP к шуму в данных. Но всё же модель смогла сохранить высокие значения Precision, Recall и *F*1-score, что указывает на ее способность сохранять точность предсказаний даже в условиях усложненной задачи. Добавление шума и увеличение кросс-валидации усложнили задачу для модели, это привело к небольшому снижению производительности. Однако показатели Precision, Recall и *F*1-score остаются высокими (0,97), что говорит о хорошей способности модели справляться с задачей классификации сетевого трафика и выявления атак даже при наличии шумных данных. Модель сохраняет точность и устойчивость, несмотря на усложнение задачи.

Однако следует отметить, что добавленный шум с фиксированными параметрами (коэффициент шума 0,7 и нормальное распределение) может не в полной мере отражать реальные условия работы с сетевым трафиком. Реальные данные могут содержать более сложные виды шума, такие как выбросы или коррелированные отклонения, которые трудно смоделировать с помощью стандартного гауссова шума. Поэтому для более точной оценки устойчивости модели необходимо тестировать ее на разнообразных данных или использовать более сложные методы генерации шума, чтобы лучше имитировать реальные

условия, например, такие как Outliers – добавление выбросов. Для этого к случайным данным можно добавить редкие значения с большим отклонением (например, используя распределение Коши) [29] или использовать шумы с временной зависимостью [30].

Перейдем к последнему критерию оценки работы модели MLP – ROC-кривой. ROC-кривые на двух изображениях показывают результаты работы многослойного перцептрона (MLP) при разных условиях – с обычными параметрами, с добавлением шума и использованием кросс-валидации.

На рис. 7 представлен вариант 1 – модель без шума и без кросс-валидации.

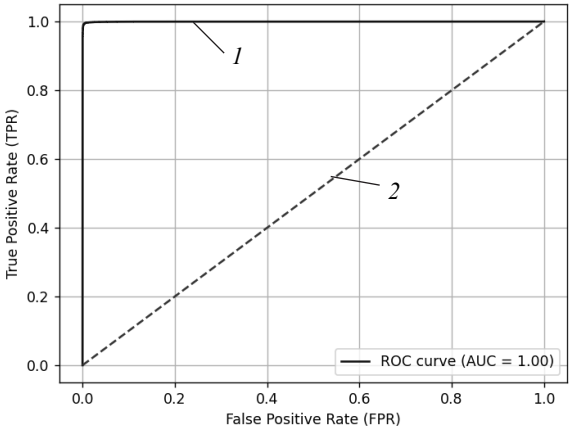

Рис. 7. ROC-кривая модели MLP, обученной на датасете NSL-KDD без добавления шумов и валидации

Сплошная линия – это кривая ROC, представляющая зависимость доли истинно положительных результатов (True Positive Rate, TPR) от доли ложноположительных результатов (False Positive Rate, FPR) при различных порогах классификации. Чем ближе линия *1* к верхнему левому углу графика, тем лучше работает модель. В данном случае линия *1* касается верхнего края графика, что указывает на идеальную классификацию.

Пунктирная линия – линия случайной классификации, при которой значения TPR и FPR равны. Эта линия служит базовым ориентиром: если кривая модели лежит ниже линии случайной классификации или на ее уровне, то модель классифицирует не лучше, чем случайное угадывание (AUC = 0,5). Модель считается полезной, если ее ROC-кривая находится выше этой линии

(AUC > 0,5). На графике линия случайной классификации проходит по диагонали от координаты (0, 0) до (1, 1).

Рассмотрим параметры графика:

- ROC-кривая практически идеально совпадает с верхней границей графика, что говорит о безошибочной классификации: все атаки и нормальные соединения классифицированы правильно;
- ROC-AUC = 1,00 означает, что модель безошибочно классифицирует данные на тестовом наборе. Кривая показывает, что на любых порогах классификации модель не допускает ложных положительных или ложных отрицательных срабатываний.

На рис. 8 представлен вариант 2 – модель с шумом и кросс-валидацией.

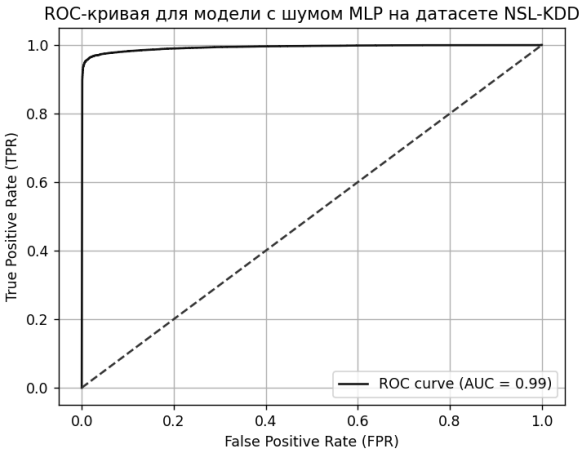

Рис. 8. ROC-кривая модели MLP, обученной на датасете NSL-KDD с добавлением шумов и валидации

ROC-кривая остается близкой к верхней границе графика, однако незначительное отклонение от идеала указывает на то, что модель допустила небольшое количество ошибок при классификации данных.

Добавление шума и использование кросс-валидации усложнило задачу для модели, и это повлияло на производительность. Однако модель по-прежнему демонстрирует высокие результаты и сохраняет почти идеальную способность классификации сетевого трафика.

Многослойный перцептрон (MLP) показал высокую эффективность на датасете NSL-KDD даже с применением шумов и кросс-валидации, достигнув почти идеальных результатов с точностью 96,94 % и значением ROC-AUC, равным 0,99. Модель успешно классифицировала сетевой трафик, выявляя кибератаки. Однако в реальных условиях результаты могут измениться: реальный трафик более разнообразен и содержит неструктурированный шум, а также новые типы атак, не представленные в тестовых данных. Это может привести к снижению производительности модели. Для повышения устойчивости модели к реальным данным рекомендуется использовать методы поиска гиперпараметров, такие как Grid Search [31], другие типы регуляризации, например $L1$-регуляризация или Dropout, и увеличение фактора шума или синтетических данных с помощью генеративных моделей, таких как GAN. Также в дальнейшем можно сравнить производительность MLP с другими моделями, чтобы выбрать оптимальную архитектуру для классификации трафика в реальных условиях.

ВЫВОДЫ

В настоящей работе было рассмотрено применение многослойного перцептрона (MLP) для классификации сетевого трафика и обнаружения киберугроз на основе датасета NSL-KDD. Исследование показало, что многослойный перцептрон является эффективным инструментом для задачи классификации трафика, позволяющим выявлять кибератаки с высокой точностью. Модель продемонстрировала отличные результаты по ключевым метрикам даже в условиях добавления шума и кросс-валидации: точность классификации составила 96,94 %, а значение ROC-AUC достигло 0,99, что указывает на практически идеальную способность модели различать нормальные соединения и атаки.

Для результатов были выполнены следующие этапы предобработки данных. Во-первых, категориальные признаки, такие как тип протокола и служба, были закодированы с использованием метода One-Hot Encoding, что позволило преобразовать текстовые данные в числовые, удобные для обработки моделью. Во-вторых, для решения проблемы дисбаланса классов был применен метод SMOTE, который увеличил количество классов атак и позволил модели лучше обучаться на примерах атакующих соединений. В случае отсутствия метода One-Hot Encoding классификация категориальных данных, таких как типы протоколов или службы, становилась бы невозможной, так как не было бы преобразования в числовой формат, с которым работает MLP,

а без метода SMOTE модель была бы менее способна обучиться на примерах атакующего трафика, так как у нее было бы мало примеров атак для анализа.

Архитектура модели MLP включала два скрытых слоя (50 и 25 нейронов), что обеспечило баланс между сложностью модели и ее способностью обобщать на новые данные.

Несмотря на достигнутые результаты, есть несколько направлений для дальнейших улучшений модели. Прежде всего следует разработать требования к набору данных, чтобы убедиться в устойчивости модели к неизвестным или ранее не встречавшимся атакам. Для повышения ее устойчивости в реальных условиях целесообразно применить методы оптимизации гиперпараметров, такие как Grid Search, а также рассмотреть другие методы регуляризации, например $L1$-регуляризацию или Dropout. Кроме того, увеличение фактора шума и использование синтетических данных, сгенерированных с помощью моделей, таких как GAN, может помочь улучшить способность модели к обобщению. В перспективе целесообразно сравнить производительность MLP с другими моделями, чтобы определить оптимальную архитектуру для классификации сетевого трафика в условиях реальных киберугроз.

СПИСОК ЛИТЕРАТУРЫ

1. *Sommer R., Paxson V.* Outside the closed world: on using machine learning for network intrusion detection // 2010 IEEE Symposium on Security and Privacy. – IEEE, 2010. – P. 305–316. – DOI: 10.1109/SP.2010.25.

2. *Roesch M.* Snort: lightweight intrusion detection for networks // Proceedings of the 13th USENIX Conference on System Administration (LISA '99). – USENIX, 1999. – P. 229–238.

3. Fuzziness based semi-supervised learning approach for intrusion detection system / R.A.R. Ashfaq, X.Z. Wang, J.Z. Huang, H. Abbas, Y.L. He // Information Sciences. – 2017. – Vol. 378. – P. 484–497.

4. A deep learning approach to network intrusion detection / N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi // IEEE Transactions on Emerging Topics in Computational Intelligence. – 2018. – Vol. 2 (1). – P. 41–50.

5. Intrusion detection system: a comprehensive review / H.J. Liao, C.H.R. Lin, Y.C. Lin, K.Y. Tung // Journal of Network and Computer Applications. – 2013. – Vol. 36 (1). – P. 16–24.

6. *Bhuyan M.H., Bhattacharyya D.K., Kalita J.K.* Network anomaly detection: methods, systems and tools // IEEE Communications Surveys & Tutorials. – 2014. – Vol. 16 (1). – P. 303–336.

7. A deep learning approach for network intrusion detection system / A. Javaid, Q. Niyaz, W. Sun, M. Alam // Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies. – ACM, 2016. – P. 21–26.

8. *Vinayakumar R., Soman K.P., Poornachandran P.* Applying deep learning approaches for network traffic classification and intrusion detection // 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). – IEEE, 2017. – P. 1222–1228.

9. *Aggarwal C.C.* Neural networks and deep learning: a textbook. – Springer, 2018.

10. *Goodfellow I., Bengio Y., Courville A.* Deep learning. – MIT Press, 2016.

11. *Akbar K.A.C., Varma P.M.* Intrusion detection system based on multi-layer perceptron neural networks // International Journal of Computer Applications. – 2012. – Vol. 52 (7). – P. 25–30.

12. *Ahmed M.S.S.* Intrusion detection system using MLP neural network with packet statistical features // Journal of Communications Software and Systems. – 2019. – Vol. 15 (3). – P. 267–274.

13. *Goodfellow I., Bengio Y., Courville A*. Deep learning. – MIT Press, 2016.

14. *Nielsen M.A.* Neural networks and deep learning. – Determination Press, 2015.

15. Deep learning approach for intelligent intrusion detection system / R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, S. Venkatraman // IEEE Access. – 2019. – Vol. 7. – P. 41525–41550.

16. Method of intrusion detection using deep neural network / Y. Kim, J. Lee, Y. Kim, H.K. Kim // 2017 International Conference on Big Data and Smart Computing (BigComp). – IEEE, 2018. – P. 313–316.

17. *Li Y., Wang Y.* A hybrid malicious code detection method based on deep learning // International Journal of Security and Its Applications. – 2018. – Vol. 12 (2). – P. 71–82.

18. Network traffic classifier with convolutional and recurrent neural networks for internet of things / M. López-Martín, B. Carro, A. Sánchez-Esguevillas, J. Lloret // IEEE Access. – 2017. – Vol. 5. – P. 18042–18050.

19. *Chiu W.Y., Tsai Y.H., Li M.H.* Improving network intrusion detection by the time-related features // 2015 IEEE International Conference on Applied System Innovation (ICASI). – IEEE, 2015. – P. 997–1000.

20. *Bishop C.M.* Pattern recognition and machine learning. – Springer, 2006.

21. A detailed analysis of the KDD CUP 99 data set / M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani // Proceedings of the 2009 IEEE Symposium on Computa-

tional Intelligence in Security and Defense Applications. – IEEE, 2009. – DOI: 10.1109/CISDA.2009.5356528.

22. *Han J., Kamber M., Pei J.* Data mining: concepts and techniques. – 3rd ed. – Morgan Kaufmann, 2012.

23. SMOTE: synthetic minority over-sampling technique / N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer // Journal of Artificial Intelligence Research. – 2002. – Vol. 16. – P. 321–357.

24. *Zhang J., Li W., Liu Z.* Enhancing intrusion detection using noise injection in deep neural networks // Security and Communication Networks. – 2018. – Art. 6725018.

25. *Kohavi R.* A study of cross-validation and bootstrap for accuracy estimation and model selection // Artificial Intelligence. – 1995. – Vol. 14 (2). – P. 1137–1143.

26. *Bishop C.M.* Training with noise is equivalent to Tikhonov regularization // Neural Computation. – 1995. – Vol. 7 (1). – P. 108–116.

27. *Aggarwal C.C.* Outlier analysis. – 2nd ed. – Springer, 2016.

28. *Box G.E., Jenkins G. M., Reinsel G.C.* Time series analysis: forecasting and control. – 5th ed. – Wiley, 2015.

29. *Hsu C.W., Chang C.C., Lin C.J.* A practical guide to support vector classification. Technical Report. – National Taiwan University, 2010.

30. *Ng A.Y.* Feature selection, L1 vs. L2 regularization, and rotational invariance // Proceedings of the 21st International Conference on Machine Learning (ICML). – ACM, 2004. – P. 615–622.

31. Generative adversarial nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Advances in Neural Information Processing Systems (NeurIPS). – Montreal, 2014.

Подсевалов Артем Георгиевич, аспирант кафедры защиты информации Новосибирского государственного технического университета. Основное направление научных исследований: управление уязвимостями, мониторинг событий информационной безопасности. E-mail: podsevalov.2019@stud.nstu.ru

Киселев Максим Алексеевич, аспирант кафедры защиты информации Новосибирского государственного технического университета. Основное направление научных исследований: кибербезопасность, мониторинг событий безопасности, экспертные системы. E-mail: m.kiselev.2019@stud.nstu.ru

Иванов Андрей Валерьевич, кандидат технических наук, доцент, заведующий кафедрой защиты информации Новосибирского государственного тех-

нического университета. Область научных интересов: информационная безопасность, кибербезопасность. E-mail: andrej.ivanov@corp.nstu.ru

DOI: 10.17212/2782-2230-2024-4-37-65

Application of Multilayer Perceptron (MLP) neural network for detection and classification of cyber threats in network traffic*

A.G. Podsevalov[1], M.A. Kiselev[2], A.V. Ivanov[3]

[1] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, graduate student of Information Security Department. E-mail: podsevalov.2019@stud.nstu.ru.*
[2] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, graduate student of Information Security Department. E-mail: m.kiselev.2019@stud.nstu.ru.*
[3] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, PhD in Technology, head of the Information Security Department. E-mail: andrej.ivanov@corp.nstu.ru.*

This article examines the application of a multilayer perceptron (MLP) for network traffic classification aimed at detecting cyber threats. The model was trained on the NSL-KDD dataset, a standard dataset widely used in research for attack detection tasks. During the experiments, data preprocessing was conducted, including encoding of categorical features and class balancing using the SMOTE method to address the imbalance between normal and malicious traffic. The results demonstrated high classification accuracy of 96,64 %, even under noise conditions and 10-fold cross-validation, which confirms the reliability of the proposed approach. The article presents performance metrics such as accuracy, recall, and F1-score, which can serve as a foundation for further research and optimization of machine learning models to enhance network security.

Keywords: information security, cyber threats, machine learning, MLP, NSL-KDD, ROC-AUC, SMOTE, cross-validation, One-Hot Encoding

* *Received 20 November 2024.*

REFERENCES

1. Sommer R., Paxson V. Outside the closed world: on using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 305–316. DOI: 10.1109/SP.2010.25.

2. Roesch M. Snort: lightweight intrusion detection for networks. *Proceedings of the 13th USENIX Conference on System Administration (LISA '99)*, 1999, pp. 229–238.

3. Ashfaq R.A.R., Wang X.Z., Huang J.Z., Abbas H., He Y.L. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 2017, vol. 378, pp. 484–497.

4. Shone N., Ngoc T.N., Phai V.D., Shi Q. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018, vol. 2 (1), pp. 41–50.

5. Liao H.J., Lin C.H.R., Lin Y.C., Tung K.Y. Intrusion detection system: a comprehensive review. *Journal of Network and Computer Applications*, 2013, vol. 36 (1), pp. 16–24.

6. Bhuyan M.H., Bhattacharyya D.K., Kalita J.K. Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 2014, vol. 16 (1), pp. 303–336.

7. Javaid A., Niyaz Q., Sun W., Alam M. A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies*. ACM, 2016, pp. 21–26.

8. Vinayakumar R., Soman K.P., Poornachandran P. Applying deep learning approaches for network traffic classification and intrusion detection. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

9. Aggarwal C.C. *Neural networks and deep learning*. Springer, 2018.

10. Goodfellow I., Bengio Y., Courville A. *Deep learning*. MIT Press, 2016.

11. Akbar K.A.C., Varma P.M. Intrusion detection system based on multi-layer perceptron neural networks. *International Journal of Computer Applications*, 2012, vol. 52 (7), pp. 25–30.

12. Ahmed M.S.S. Intrusion detection system using MLP neural network with packet statistical features. *Journal of Communications Software and Systems*, 2019, vol. 15 (3), pp. 267–274.

13. Goodfellow I., Bengio Y., Courville A. *Deep learning*. MIT Press, 2016.

14. Nielsen M.A. *Neural networks and deep learning*. Determination Press, 2015.

15. Vinayakumar R., Alazab M., Soman K.P., Poornachandran P., Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 2019, vol. 7, pp. 41525–41550.

16. Kim Y., Lee J., Kim Y., Kim H.K. Method of intrusion detection using deep neural network. *2017 International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2018, pp. 313–316.

17. Li Y., Wang Y. A hybrid malicious code detection method based on deep learning. *International Journal of Security and Its Applications*, 2018, vol. 12 (2), pp. 71–82.

18. López-Martín M., Carro B., Sánchez-Esguevillas A., Lloret J. Network traffic classifier with convolutional and recurrent neural networks for internet of things. *IEEE Access*, 2017, vol. 5, pp. 18042–18050.

19. Chiu W.Y., Tsai Y.H., Li M.H.. Improving network intrusion detection by the time-related features. *2015 IEEE International Conference on Applied System Innovation (ICASI)*, 2015, pp. 997–1000.

20. Bishop C.M. *Pattern recognition and machine learning*. Springer, 2006.

21. Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set. *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications*, 2009. DOI: 10.1109/CISDA.2009.5356528.

22. Han J., Kamber M., Pei J. *Data mining: concepts and techniques*. 3rd ed. Morgan Kaufmann, 2012.

23. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002, vol. 16, pp. 321–357.

24. Zhang J., Li W., Liu Z. Enhancing intrusion detection using noise injection in deep neural networks. *Security and Communication Networks*, 2018, art. 6725018.

25. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Artificial Intelligence*, 1995, vol. 14 (2), pp. 1137–1143.

26. Bishop C.M. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 1995, vol. 7 (1), pp. 108–116.

27. Aggarwal C.C. *Outlier analysis*. 2nd ed. Springer, 2016.

28. Box G.E., Jenkins G. M., Reinsel G.C. *Time series analysis: forecasting and control*. 5th ed. Wiley, 2015.

29. Hsu C.W., Chang C.C., Lin C.J. *A practical guide to support vector classification*. Technical Report. National Taiwan University, 2010.

30. Ng A.Y. Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the 21st International Conference on Machine Learning (ICML)*. ACM, 2004, pp. 615–622.

31. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, 2014.

Для цитирования:

Подсевалов А.Г., Киселев М.А., Иванов А.В. Применение нейронной сети Multilayer Perceptron (MLP) для обнаружения и классификации киберугроз в сетевом трафике // Безопасность цифровых технологий. – 2024. – № 4 (115). – С. 37–65. – DOI: 10.17212/2782-2230-2024-4-37-65.

For citation:

Podsevalov A.G., Kiselev M.A., Ivanov A.V. Primenenie neironnoi seti Multilayer Perceptron (MLP) dlya obnaruzheniya i klassifikatsii kiberugroz v setevom trafike [Application of Multilayer Perceptron (MLP) neural network for detection and classification of cyber threats in network traffic]. *Bezopasnost' tsifrovykh tekhnologii = Digital Technology Security*, 2024, no. 4 (115), pp. 37–65. DOI: 10.17212/2782-2230-2024-4-37-65.