Научная статья
УДК 004.8
DOI:10.31854/1813-324X-2023-9-4-97-113

Многозначная классификация компьютерных атак с использованием искусственных нейронных сетей с множественным выходом

Олег Иванович Шелухин, sheluhin@mail.ru
Дмитрий Игоревич Раковский ✉, Prophet_alpha@mail.ru

Московский технический университет связи и информатики,
Москва, 111024, Российская Федерация

Аннотация: *Современные компьютерные сети (КС), имея сложную и часто гетерогенную структуру, порождают большие объемы многомерных многозначных данных. Учет информации о многозначности экспериментальных данных (ЭД) может повысить эффективность решения целого ряда задач информационной безопасности: от профилирования КС до обнаружения и предотвращения компьютерных атак на КС. Целью работы является разработка многозначной архитектуры искусственной нейронной сети (ИНС) для обнаружения и классификации компьютерных атак в многозначных ЭД, и ее сравнительный анализ с известными аналогами по бинарным метрикам оценки качества классификации. Рассмотрена формализация ИНС в терминах матричной алгебры, позволяющая учитывать случай многозначной классификации и новая архитектура ИНС с множественным выходом с использованием предложенной формализации. Достоинством предложенной формализации является лаконичность ряда записей, ассоциированных с рабочим режимом работы ИНС и режимом обучения. Предложенная архитектура ИНС позволяет решать задачи обнаружения и классификации многозначных компьютерных атак в среднем на 5 % эффективнее известных аналогов. Наблюдаемый выигрыш обусловлен учетом многозначных закономерностей между классовыми метками на этапе обучения за счет использования общего первого слоя. Достоинствами предложенной архитектуры ИНС является масштабируемость к любому числу классовых меток и быстрая сходимость.*

Ключевые слова: *информационная безопасность, многозначная классификация, компьютерная сеть, компьютерная атака, нейронные сети, глубокие нейронные сети, многозначные нейронные сети*

Ссылка для цитирования: Шелухин О.И., Раковский Д.И. Многозначная классификация компьютерных атак с использованием искусственных нейронных сетей с множественным выходом // Труды учебных заведений связи. 2023. Т. 9. № 4. С. 97–113. DOI:10.31854/1813-324X-2023-9-4-97-113

Multivalued Classification of Computer Attacks Using Artificial Neural Networks with Multiple Outputs

Oleg Shelukhin, sheluhin@mail.ru
Dmitry Rakovsky ✉, Prophet_alpha@mail.ru

[1]Moscow Technical University of Communications and Informatics,
Moscow, 111024, Russian Federation

Abstract: *Modern computer networks (CN), having a complex and often heterogeneous structure, generate large volumes of multi-dimensional multi-label data. Accounting for information about multi-label experimental data (ED) can improve the efficiency of solving a number of information security problems: from CN profiling to detecting and preventing computer attacks on CN. The aim of the work is to develop a multi-label artificial neural network (ANN) architecture for detecting and classifying computer attacks in multi-label ED, and its comparative analysis with known analogues in terms of binary metrics for assessing the quality of classification. A formalization of ANN in terms of matrix algebra is proposed, which allows taking into account the case of multi-label classification and the new architecture of ANN with multiple output using the proposed formalization. The advantage of the proposed formalization is the conciseness of a number of entries associated with the ANN operating mode and learning mode. Proposed architecture allows solving the problems of detecting and classifying multi-label computer attacks, on average, 5% more efficiently than known analogues. The observed gain is due to taking into account multi-label patterns between class labels at the training stage through the use of a common first layer. The advantages of the proposed ANN architecture are scalability to any number of class labels and fast convergence.*

Keywords: *information security, Multi-label classification, computer network, computer attack, neural networks, deep neural networks, Multi-label neural networks*

For citation: Shelukhin O., Rakovsky D. Multivalued Classification of Computer Attacks Using Artificial Neural Networks with Multiple Outputs. *Proceedings of Telecommun. Univ.* 2023;9(4):97–113. DOI:10.31854/1813-324X-2023-9-4-97-113

Введение

Современные компьютерные сети (КС), имея сложную и часто гетерогенную структуру [1], порождают большие объемы многомерных данных. Как показывают исследования, КС порождает многозначные экспериментальные данные (ЭД), одновременно ассоциированные сразу с несколькими классовыми метками [2]. Учет информации о многозначности ЭД может повысить эффективность решения целого ряда задач: профилирование и прогнозирование состояний КС «в будущем» [3, 4]; прогнозирование редких аномальных состояний КС [5]; обнаружение и предотвращение компьютерных атак на КС [6]. Применение многозначного анализа позволяет повысить эффективность решения задач информационной безопасности (ИБ), решение которых классическими «однозначными» способами затруднено. Это, например, анализ инцидентов ИБ [7]; маркировка подозрительного сетевого трафика [8]; классификация зашифрованного сетевого трафика [9].

Одним из актуальных инструментов решения задач многозначной классификации являются искусственные нейронные сети, ИНС (ANN, *аббр. от англ.* Artificial Neural Network). Для задач многозначной классификации используют нейронные сети с глубоким обучением, которые могут извлекать из данных более абстрактные признаки и моделировать более сложные нелинейные отношения между ними [10]. Для исследования подобных процессов могут использоваться различные архитектуры ИНС (по отдельности или в комбинации): полносвязная – многослойный персептрон (MLP, *аббр. от англ*. Multi-Layer Perceptron); сверточная нейронная сеть (CNN, *аббр. от англ.* Convolutional Neural Networks); рекуррентная нейронная сеть (RNN, *аббр. от англ.* Recurrent Neural Networks); нейронные сети типа «трансформер» (Transformer) [11, 12]. Известны спайковые нейронные сети и их приложения для анализа изображений, что может быть полезно для решения ряда задач информационной безопасности [13].

Целью работы является разработка многозначной архитектуры ИНС для обнаружения и классификации компьютерных атак в многозначных ЭД, и ее сравнительный анализ с известными аналогами по бинарным метрикам оценки качества классификации.

Формализация задачи многозначной классификации компьютерных атак

КС можно представить, как множество из M упорядоченных наборов значений ее дискретно изменяющихся атрибутов («исторических данных»):

$$\begin{gathered} \mathrm{A} \subseteq \mathrm{A}_{\text{перв}} \cup \mathrm{A}_{\text{втор}} = \\ = \left(A_{\text{перв } 1} \times A_{\text{перв } 2} \times \ldots \times A_{\text{перв } len_1}\right) \cup \\ \cup \left(A_{\text{втор } 1} \times A_{\text{втор } 2} \times \ldots \times A_{\text{втор } len_2}\right) \end{gathered} \quad (1)$$

Каждый член записи (1) является атрибутом КС и может быть представлен в виде вектора-столбца:

$$\mathrm{A}(\ , m) = \left(a_{1m}, a_{2m}, \ \ldots \ , a_{Nm}; m = \overline{1, M}, n = \right.$$
$$\left. = \ \overline{1, N}, M = len_1 + len_2 \ \right).$$

Атрибуты КС могут подразделяться на два типа – первичные и вторичные:

$$\left(A_{\text{перв } k_1}; k_1 = \overline{1, len_1}\right),$$

$$\left(A_{\text{втор } k_2}; k_2 = \overline{1, len_2}\right).$$

Первичные атрибуты получают непосредственно от системных датчиков, установленных внутри КС. Вторичные атрибуты получают в результате обработки первичных атрибутов. Примерами вто-

ричных атрибутов могут быть, например, среднее время задержки сигнала в КС, количество потерянных пакетов в КС для конкретного хоста и пр.

Как правило, исторические данные, преобразованные к табличному виду посредством ряда манипуляций [2, 6], могут быть представлены в виде таблицы размером M столбцов на N строк:

$$D_{NM} = \\ = \{(\mathrm{A}(n,\ ), set_n); \mathrm{A} = (a_{nm}), m = \overline{1,M}, n = \overline{1,N}\}, \quad (2)$$

где n-й строке значений атрибутов записи $\mathrm{A}(n,\ )$ ставится в соответствие множество меток set_n. Множество меток set_n ассоциируется с решающими правилами в контексте задачи. В задачах информационной безопасности метки могут соответствовать категориальным маркерам, связанным с профилем функционирования КС (состояния КС – метки, как правило, связанные с вторичными атрибутами) или с возникновением (отсутствием) определенной компьютерной атаки.

Предполагается, что множество меток, ассоциированных с записью ЭД, многозначно – т. е. одной записи может соответствовать одновременно несколько классовых меток.

Извлечем ряд таких множеств в отдельное упорядоченное множество по правилу:

$$f: D_{NM} \to L_n; L_n = (set_n; n = \overline{1,N}), \quad (3)$$

где D_{NM} – табличное представление размеченных экспериментальных данных; L_n – множеством меток set_n.

«Алфавит», образованный уникальными элементами множества L_n, сведем в отдельное множество S:

$$S = \bigcup_{n=1}^{N} set_n, \quad (4)$$

в котором объединены все элементы L_n.

Поскольку S не является мультимножеством, повторяющиеся элементы исключаются, что позволяет получить все уникальные классовые метки, содержащиеся в «исторических данных».

Пусть немаркированные, «новые» данные одинаковой размерности с «историческими данными» (2), представлены в виде:

$$\hat{D}_{\hat{N}M} = \{\hat{\mathrm{A}}(\hat{n},\ ); \hat{\mathrm{A}} = (\hat{a}_{\hat{n}m}), m = \overline{1,M}, \hat{n} = \overline{1,\hat{N}}\}, \quad (5)$$

где $\hat{N} \in \mathbb{N}$ – количество неразмеченных данных; $\hat{\mathrm{A}}(\hat{n},\ )$ – $\hat{n}$-я строка в наборе неразмеченных экспериментальных данных.

Под классификацией немаркированной записи будем понимать процесс отображения немаркированной $\hat{n}$-й записи (вектора-строки) $\hat{\mathrm{A}}(\hat{n},\ )$ в соответствующий набор классовых меток $set_{\hat{n}}$. Процесс маркировки $\hat{\mathrm{A}}(\hat{n},\ )$ может быть выполнен посредством аппроксимации $\hat{\mathrm{A}}(\hat{n},\ )$ по некоторой метрике ф, ассоциированной с истинным набором многозначных классовых меток для $\hat{\mathrm{A}}(\hat{n},\ )$ – $set_{\hat{n}}^{true}$. Смысл «метрики ф» определяется подходом к решению задачи классификации. Как правило, постановка задачи классификации формируется на базе аксиоматики Колмогорова [14]. В рамках такой постановки под решением задачи классификации понимается построение алгоритма $\mathrm{ALG}_1\left(\hat{\mathrm{A}}(\hat{n},\ )\right)$, вероятность ошибочной классификации которого может быть формализована соотношением [15]:

$$P\left(\mathrm{ALG}_1\left(\hat{\mathrm{A}}(\hat{n},\ )\right) = set_{\hat{n}},\quad set_{\hat{n}} \neq set_{\hat{n}}^{true}\right) \to \min.$$

Метрикой $ф_1 \in [0,1]$ в данном случае выступает вероятность ошибочной классификации. Однако использование аксиоматики Колмогорова сопряжено с некоторыми ограничениями [16], такими, например, как необходимость знания априорной вероятности распределения классовых меток, что не всегда выполняется для реальных, сложных компьютерных систем и сетей.

Для преодоления указанных ограничений задача отображения $\hat{\mathrm{A}}(\hat{n},\ ) \to set_{\hat{n}}^{true}$ может быть выражена в понятиях аналитической геометрии (т. н. кластерный анализ и ассоциированная с ним задача многомерной классификации) или теории множеств. При формулировании задачи классификации в понятиях аналитической геометрии необходимо разбить анализируемую совокупность ЭД на сравнительно небольшое число, в определенном смысле однородных групп или классов с использованием меры сходства (расстояние, близость) $ф_2$ [17].

В результате задача классификации может быть сформулирована следующим образом [18].

Пусть дана неразмеченная запись ЭД в виде множества $\hat{\mathrm{A}}(\hat{n},\ )$ и множество меток $set_{\hat{n}}$, ассоциированное с $\hat{\mathrm{A}}(\hat{n},\ )$. Тогда результатом классификации будет точечно-множественное отображение $\mu: \hat{\mathrm{A}}(\hat{n},\ ) \to set_{\hat{n}}^{true}$, такое, что при пересечении множеств предсказанных $set_{\hat{n}}$ и истинных $set_{\hat{n}}^{true}$ меток, результирующее множество будет равным истинному – $set_{\hat{n}} \cap set_{\hat{n}}^{true} = set_{\hat{n}}^{true}$.

Рассмотренные варианты постановки задач классификации могут быть обобщены и на смежные области – например, на задачу прогнозирования. Отметим, что в реальных задачах набор истинных меток $set_{\hat{n}}^{true}$, ассоциированный с поступающими на вход неразмеченными ЭД, часто неизвестен – или становится известен спустя продолжительное время наблюдений, что неприемлемо для задач ИБ. Поэтому для оценки качества классификации алгоритма обычно предполагают разделение известной, размеченной, выборки ЭД D_{NM} на три подмножества: обучающую, валидационную и тестовую [19].

Валидационная выборка предъявляется в качестве эталонной для оценки качества классификации во время обучения алгоритма (например, при обучении ИНС, такая выборка предъявляется по окончании каждой эпохи обучения сети).

Тестовая выборка имитирует поступление неразмеченных ЭД на вход к алгоритму классификации в условиях, приближенных к реальным. Алгоритм, как правило, не имеет информации о записях, ассоциированных с приведенной выборкой. При этом результаты алгоритма ($set_{\hat{n}}$) – сравниваются с истинными метками, ассоциированными с тестовой выборкой ($set_{\hat{n}}^{true}$).

Таким образом достигается апробация алгоритма на «неразмеченных данных», информация о которых неизвестна алгоритму классификации.

Структура ИНС для многозначной классификации

Рассмотрим полносвязную ИНС, состоящую из LC-слоев (*аббр. от англ.* LayerCount), каждый слой которой описывается NCAL- или *ncal*-нейронами (*аббр. от англ.* Neurons Count At Layer).

Каждый отдельный нейрон может быть описан правилом преобразования входного вектора $\overrightarrow{Input}_{ncal_{lc},lc}$ (ассоциированного с *lc*-м слоем ИНС и $ncal_{lc}$-м нейроном) в выходному вектору $\overrightarrow{Output}_{ncal_{lc},lc}$:

$$AF_{ncal_{lc},lc}\left(net_{ncal_{lc},lc}\right) \rightarrow \overrightarrow{Output}_{ncal_{lc},lc}, \qquad (6)$$

где $net_{ncal_{lc},lc} = \left\langle \overrightarrow{Input}_{ncal_{lc},lc}, \vec{W}_{ncal_{lc},lc} \right\rangle + w_{\text{смещ.}ncal_{lc},lc}$; lc – номер слоя, в котором расположен нейрон, $lc = \overline{1, LC}$; $ncal_{lc}$ – номер нейрона на слое $lc, ncal_{lc} = \overline{1, NCAL_{lc}}; NCAL_{lc} \in NCAL\_library$; $\overrightarrow{Input}_{ncal_{lc},lc}$ – входной вектор, поступающий на вход нейрона; $\overrightarrow{W_{lc,ncal_{lc}}}$ – вектор весов синапсов (входных связей) нейрона; $\left\langle \overrightarrow{Input}_{ncal_{lc},lc}, \vec{W}_{ncal_{lc},lc} \right\rangle$ – скалярное произведение между входным вектором и вектором весов, скаляр, $net_{ncal_{lc},lc}$; $w_{\text{смещ. } ncal_{lc},lc}$ – вход смещения, ассоциированный с нейроном; $\overrightarrow{Output}_{ncal_{lc},lc}$ – выходной вектор, полученный после прохождения $net_{ncal_{lc},lc}$ через функцию активации $AF_{ncal_{lc},lc}\left(net_{ncal_{lc},lc}\right)$.

Для учета различия количества нейронов в каждом слое дополнительно вводится множество $NCAL\_library$, содержащее информацию о размере каждого слоя ИНС $NCAL_{lc} \in NCAL\_library$.

В общем случае $\overrightarrow{Output}_{ncal_{lc},lc}$ является упорядоченным набором скаляров (вектором), формируемым функцией активации $AF_{ncal_{lc},lc}\left(net_{ncal_{lc},lc}\right)$ и архитектурой слоя. Выходной вектор может как вырождаться в скалярную величину (состоять из одного элемента), так и трансформироваться в матрицу или тензор. Как правило, на уровне скрытого слоя многослойной полносвязной ИНС, при наличии входных данных, сформированных в виде двухмерной табличной записи (2), применяются функции активации, возвращающие скалярную величину сигмоидального вида *ReLU* и др. Многомерные функции активации задействованы во многих архитектурах: *Long Short Term Memory, Gated Recurrent Unit, Transformer*, а также в *Softmax* [20].

Отдельный полносвязный *lc*-й слой, состоящий из NCAL-нейронов, может быть записан в виде вектора-столбца матрицы, ассоциированный с полносвязной ИНС (*NN, аббр. от англ.* Neural Network):

$$NN(\ , lc) = \begin{pmatrix} AF_{1,lc}\left(net_{1,lc}\right) \\ AF_{2,lc}\left(net_{2,lc}\right) \\ \dots \\ AF_{NCAL_{lc},lc}\left(net_{NCAL_{lc},lc}\right) \end{pmatrix}, \qquad (7)$$

где $net_{ncal_{lc},lc} = \left\langle NN(\ , lc-1), \vec{W}_{ncal_{lc},lc} \right\rangle + w_{\text{смещ. } ncal_{lc},lc}$, $ncal_{lc} = \overline{1, NCAL_{lc}}; NCAL_{lc} \in NCAL\_library$.

Скалярное произведение $\left\langle NN(\ , lc-1), \vec{W}_{ncal_{lc},lc} \right\rangle$ выполняется между вектором-столбцом предыдущего слоя ИНС и вектором весовых коэффициентов синапсов, ассоциированных с определенным нейроном.

В результате матричная запись (7) полносвязной ИНС, состоящей из LC-слоев, может быть преобразована к выражению (8), где $net_{ncal_{lc},lc} = \left\langle NN(\ , lc-1), \vec{W}_{ncal_{lc},lc} \right\rangle + w_{\text{смещ. } ncal_{lc},lc}$ – скалярное произведение между вектором, сформированным предыдущим слоем, и вектором весов, ассоциированным с вектором весом *ncal*-го нейрона на *lc*-м слое, суммируемое со смещением $w_{\text{смещ. } ncal_{lc},lc}$; $AF_{ncal_{lc},lc}\left(net_{ncal_{lc},lc}\right) \in ActivationFunc$ – функция активации *ncal*-го нейрона на *lc*-м слое; $lc = \overline{1, LC}$, $ncal_{lc} = \overline{1, NCAL_{lc}}; NCAL_{lc} \in NCAL\_library$:

$$NN = \begin{pmatrix} AF_{1,1}\left(net_{1,1}\right) & \dots & AF_{1,lc}\left(net_{1,lc}\right) & \dots & AF_{1,LC}\left(net_{1,LC}\right) \\ AF_{2,1}\left(net_{2,1}\right) & \cdots & AF_{2,lc}\left(net_{2,lc}\right) & \cdots & AF_{2,LC}\left(net_{2,LC}\right) \\ \dots & \dots & \dots & \dots & \dots \\ AF_{NCAL_1,1}\left(net_{NCAL_1,1}\right) & \cdots & AF_{NCAL_{lc},lc}\left(net_{NCAL_{lc},lc}\right) & \cdots & AF_{NCAL_{LC},LC}\left(net_{NCAL_{LC},LC}\right) \end{pmatrix}. \qquad (8)$$

Заметим, что запись (8) описывает лишь архитектуру полносвязной ИНС. Анализ ассоциированных с (8) ее гиперпараметров будет приведен ниже.

На вход каждой функции активации подается вектор, формируемый предыдущим слоем ИНС – $NN(\ ,lc-1)$. Для первого слоя в качестве входных данных используется вектор-строка немаркированных данных – $\hat{\mathrm{A}}(\hat{n},\ )$:

$$NN(\ ,1)=\begin{pmatrix} AF_{1,1}\left(\hat{\mathrm{A}}(\hat{n},\quad)\right) \\ AF_{2,1}\left(\hat{\mathrm{A}}(\hat{n},\quad)\right) \\ \dots \\ AF_{NCAL_1,1}\left(\hat{\mathrm{A}}(\hat{n},\quad)\right) \end{pmatrix}, \qquad (9)$$

где $net_{1,ncal_1}=\left\langle\hat{\mathrm{A}}(\hat{n},\ ),\vec{W}_{ncal_{lc},lc}\right\rangle+w_{\text{смещ. }\ ncal_1,1}$; $\hat{\mathrm{A}}(\hat{n},\ )$ – строка экспериментальных данных, подаваемая на вход ИНС с целью ее дальнейшей классификации; $ncal_1=\overline{1,NCAL_1}$; $NCAL_1\in NCAL\_library$.

Последний слой ИНС ассоциирован с метками классов; его размерность определяется максимально возможным одновременным количеством меток, ассоциированных с входными данными:

$$NN(\ ,LC)=\begin{pmatrix} AF_{1,LC}\left(net_{1,LC}\right) \\ AF_{2,LC}\left(net_{2,LC}\right) \\ \dots \\ AF_{NCAL_{LC},LC}\left(net_{NCAL_{LC},LC}\right) \end{pmatrix}, \qquad (10)$$

где$net_{ncal_{LC},LC}=\left\langle\hat{\mathrm{A}}(\hat{n},\ ),\vec{W}_{ncal_{LC},LC}\right\rangle+w_{\text{смещ. }\ ncal_{LC},LC}$; $\hat{\mathrm{A}}(\hat{n},\ )$ – строка ЭД, подаваемая на вход в ИНС с целью ее дальнейшей классификации; $ncal_{LC}=\overline{=1,NCAL_{LC}}$; $NCAL_{LC}\in NCAL\_library$.

Слой (10) описывает степень принадлежности входного набора значений атрибутов КС ($\hat{\mathrm{A}}(\hat{n},\ )$) к каждому из классов. В качестве функции активации в последнем слое ИНС (10) выбирается функция принятия многозначных решений – точечно-множественная функция отображения. Поскольку в рассматриваемой задаче размер слоя определяется размерностью алфавита классовых меток $|\boldsymbol{S}|$, соответственно, $NCAL_{LC}=|S|$. Многозначная реализация последнего слоя ИНС (10) может быть сведена к однозначному посредством сокращения количества нейронов в слое до одного:

$$NN(\ ,LC)=\left(AF_{LC}(net_{LC})\right),$$

где $net_{LC}=\left\langle NN(\ ,LC-1),\vec{W}_{LC}\right\rangle+w_{\text{смещ. }LC}$.

В «классическом» случае многоклассовой или бинарной классификации, решение принимается путем голосования, например – по мажоритарному принципу (принципу большинства). Для подобных задач в целях вычисления веса каждого класса и последующего определения победителя по мажоритарному принципу ($\max\left(AF_{ncal_{LC},LC}\left(net_{ncal_{LC},LC}\right)\right)$) актуально применение семейства функций активации типа *softmax*, которые возвращают вектор весов. Следовательно, результат работы последнего слоя ИНС (10), в общем случае, многозначен [21].

На практике, полносвязные ИНС (8) редко состоят из слоев одинакового размера. В задачах классификации ИНС обычно создают по аналогии с бутылочным горлышком, когда входной слой сети всегда наиболее широкий, в то время как скрытые слои «плавно» сужаются. Выходной слой равен либо одному нейрону (задача бинарной или многоклассовой классификации), либо мощности алфавита уникальных классов (задача многозначной классификации). Для учета несимметричных архитектур предлагается придерживаться правил записи треугольных матриц: заменять все отсутствующие нейроны нулями.

Приведем пример несимметричной записи полносвязной ИНС:

$$\mathrm{NN}_{example}=\\ =\begin{pmatrix} AF_{1,1}(net_{1,1}) & AF_{1,2}(net_{1,2}) & AF_{1,3}(net_{1,3}) \\ AF_{2,1}(net_{2,1}) & AF_{2,2}(net_{2,2}) & 0 \\ AF_{3,1}(net_{3,1}) & 0 & 0 \end{pmatrix}.$$

Извлечем из матричной записи полносвязной ИНС NN (8) веса Θ, ассоциированные с каждым нейроном, с учетом того, что нейрон обладает несколькими синапсами.

Для учета веса смещения объединим вектор весов, ассоциированный с синапсами $\vec{W}_{ncal_{lc},lc}$, и вес, ассоциированный с входом смещения $w_{\text{смещ. }ncal_{lc},lc}$ в двойки $\left(\vec{W}_{ncal_{lc},lc},w_{\text{смещ. }ncal_{lc},lc}\right)$. Полученную в результате матричную запись представим в виде выражения (11), где $lc=\overline{1,LC}$; $ncal_{lc}=\overline{1,NCAL_{lc}}$; $NCAL_{lc}\in NCAL\_library$.

Опишем полносвязную ИНС, заданную архитектурой слоев и функций активации NN (8), матрицей весов Θ (11) и набором гиперпараметров (Hyperparams) в виде выражения (12):

$$\Theta=\begin{pmatrix} \left(\vec{W}_{1,1},w_{\text{смещ. }\ 1,1}\right) & \dots & \left(\vec{W}_{1,lc},w_{\text{смещ. }\ 1,lc}\right) & \dots & \left(\vec{W}_{1,LC},w_{\text{смещ. }\ 1,LC}\right) \\ \left(\vec{W}_{2,1},w_{\text{смещ. }\ 2,1}\right) & \dots & \left(\vec{W}_{2,lc},w_{\text{смещ. }\ 2,lc}\right) & \dots & \left(\vec{W}_{2,LC},w_{\text{смещ. }\ 2,LC}\right) \\ \dots & \dots & \dots & \dots & \dots \\ \left(\vec{W}_{NCAL_1,1},w_{\text{смещ. }\ NCAL_1,1}\right) & \dots & \left(\vec{W}_{NCAL_{lc},lc},w_{\text{смещ. }\ NCAL_{lc},lc}\right) & \dots & \left(\vec{W}_{NCAL_{LC},LC},w_{\text{смещ. }\ NCAL_{LC},LC,}\right) \end{pmatrix}, \qquad (11)$$

$$\text{Full Neural Network}_{NN,\Theta,Hyperparams}\left(\hat{\mathrm{A}}(\hat{n},\quad)\right)=set_{\hat{n}}. \qquad (12)$$

Все три указанных объекта являются параметрами ИНС, достаточными для полного ее описания: архитектура *NN* (8) описывает структуру ИНС; матрица весов Θ (11) описывает правила преобразования входного вектора в выходной на каждом из узлов ИНС; гиперпараметры описывают стартовые условия ИНС и правила ее обучения.

ИНС с ненулевым количеством нейронов способна к преобразованию информации. С этой целью ИНС принимает на вход немаркированную $\hat{n}$-ю запись $\hat{A}(\hat{n},\ )$ из набора неразмеченных данных $\hat{D}_{\tilde{N}M}$ и отображает ее в соответствующий набор классовых меток $set_{\hat{n}}$.

Полученная нейронная сеть описывается упорядоченным множеством слоев, каждый из которых состоит из $NCAL_{lc}$-нейронов:

$$lc = \overline{1, LC};\ \ ncal_{lc} = \overline{1, NCAL_{lc}};$$
$$NCAL_{lc} \in NCAL\_library.$$

Параметр $NCAL\_library$ учитывает разницу в количестве нейронов в каждом слое ИНС. Визуализация ИНС приведена на рисунке 1.

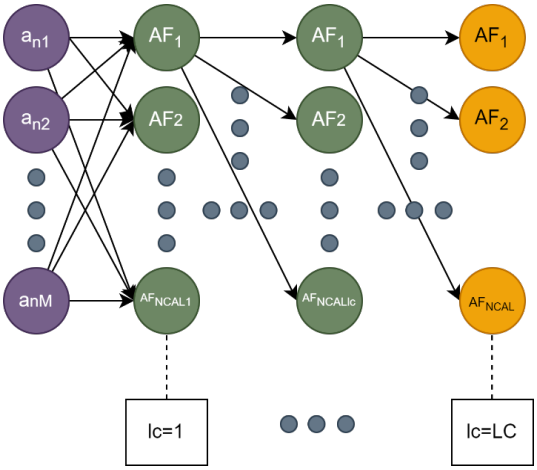

Рис. 1. Визуализация ИНС (12), с нанесенными на структуру матрицей нейронов (8)

Fig. 1. Visualization of ANN (12), with a Matrix of Neurons Applied to the Structure (8)

Гиперпараметрами ИНС, определяющими процесс ее обучения, являются: функция потерь *lossFunc*; связанная с ней функция оптимизации (весов ИНС) *optimizer*; количество эпох обучения *epoch*; количество разбиений исходных ЭД на пакеты *batchSize*.

Запишем четверку гиперпараметров как:

$$Hyperparams = \\ = (lossFunc, optimiser, epoch, batchSize). \quad (13)$$

Функция потерь $lossFunc \in LossFunc$ ассоциирована с множеством известных функций потерь (категориальная перекрестная энтропия, бинарная перекрестная энтропия и т. д. [22]. Функция оптимизации $optimiser \in OptF$ ассоциирована с множеством известных функций оптимизации (как правило, основанных на градиентном спуске [22], например – стохастический градиентный спуск, *adam* и пр.).

Режим обучения ИНС

Режим обучения ИНС (12) представляет собой итерационное предъявление элементов множества размеченных ЭД (2) на вход нейронной сети с последующей корректировкой весов нейронов (11) по функции потерь $lossFunc \in LossFunc$. Количество итераций предъявления размеченных ЭД задается с помощью переменной «количество эпох» (*epoch,* сокр. *ep*). Минимизация функции потерь (обнаружение локального (глобального) минимума *lossFunc*) производится с помощью функции оптимизации $optimiser \in OptF$.

На каждой эпохе обучения ИНС каждая запись ЭД, ассоциированная с соответствующим набором меток $\{(A(n,\ ), set_n); m = \overline{1, M}, n = \overline{1, N}\}$, подается в качестве записи на вход ИНС (11). Результирующая метка, полученная на выходе ИНС $set_n^{predict}$ – сравнивается с истинной меткой set_n, после чего вычисляется потеря по известной функции *lossFunc* и при помощи функции оптимизации *optimiser* вычисляется изменение каждого веса ИНС.

Изменение весов на каждом шаге $n = \overline{1, N}$ на эпохе *ep*, $ep = \overline{1, epoch}$, может быть записано в виде выражения (14), где $lc = \overline{1, LC};\ \ ncal_{lc} = \overline{1, NCAL_{lc}};$ $NCAL_{lc} \in NCAL\_library; n = \overline{1, N}; ep = \overline{1, epoch}.$

Корректировка веса на шаге n, $n = \overline{1, N}$ на эпохе *ep*, $ep = \overline{1, epoch}$ производится путем сложения матрицы весов ИНС, вычисленной на предыдущем шаге (n – 1), с корректировкой изменения каждого веса ИНС, вычисленной на текущем шаге для текущей записи ЭД представлено выражением (15).

$$\Delta\Theta_{n,ep} = \begin{pmatrix} (\Delta\vec{W}_{1,1}, \Delta w_{\text{смещ. } 1,1}) & \cdots & (\Delta\vec{W}_{1,lc}, \Delta w_{\text{смещ. } 1,lc}) & \cdots & (\Delta\vec{W}_{1,LC}, \Delta w_{\text{смещ. } 1,LC}) \\ (\Delta\vec{W}_{2,1}, \Delta w_{\text{смещ. } 2,1}) & \cdots & (\Delta\vec{W}_{2,lc}, \Delta w_{\text{смещ. } 2,lc}) & \cdots & (\Delta\vec{W}_{2,LC}, \Delta w_{\text{смещ. } 2,LC}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ (\Delta\vec{W}_{NCAL_1,1}, \Delta w_{\text{смещ. } NCAL_1,1}) & \cdots & (\Delta\vec{W}_{NCAL_{lc},lc}, \Delta w_{\text{смещ. } NCAL_{lc},lc}) & \cdots & (\Delta\vec{W}_{NCAL_{LC},LC}, \Delta w_{\text{смещ. } NCAL_{LC},LC,}) \end{pmatrix}. \quad (14)$$

$$\Theta_{n,ep} = \Theta_{n-1,ep} + \Delta\Theta_{n,ep}. \quad (15)$$

Наиболее актуальной стратегией вычисления корректировки для каждого веса (14) является метод обратного распространения ошибки (*backpropagation*), который заключается в вычислении дифференциала функции ошибки *lossFunc* относительно весов $\vec{W}_{ncal_{lc},lc}$ – ИНС $d\,lossFunc/d\vec{W}_{ncal_{lc},lc}$. Для корректной работы метода обратного распространения ошибки все функции активации в ИНС должны быть дифференцируемы.

Функция потерь коррелирована с метриками оценки качества классификации. Как правило, чем выше значение функции потерь, тем ниже значение метрики оценки точности – и наоборот. Поскольку ИНС с большим количеством параметров (весов синапсов) потребляет значительное количество вычислительных ресурсов, оценку эффективности решаемой задачи на этапах обучения и валидации выполняют посредством вычисления функции потерь и связанной с ней метрики оценки точности (*accuracy*, или A_m):

$$A_m = \frac{TP_m+TN_m}{TP_m+TN_m+FP_m+FN_m},$$

где m – это m-я классовая метка.

Оценка качества классификации нейронных сетей на этапе тестирования варьируется от задачи к задаче. Одним из наиболее популярных наборов метрик оценки качества многозначной классификации является основанный на Area Under Curve (AUC), площадью под Receiver Operating Characteristic (ROC).

В зависимости от методов вычисления AUC-метрики могут быть подразделены на «Один против одного» (OVO, *аббр. от англ.* One-vs-One) и на «один против всех» (OVE, *аббр. от англ.* One-vs-Everyone, или OVR, *аббр. от англ.* One-vs-Rest).

В каждом методе метрики могут быть вычислены тремя разными способами.

1) Micro – микроподход заключается в агрегации результатов классификации по каждому из M классовых меток отдельно по каждой метрике, после чего происходит вычисление итоговой метрики:

$$B_{\text{micro}} = \\ = B\left(\sum_{m=1}^{M} TP_m, \sum_{m=1}^{M} TN_m, \sum_{m=1}^{M} FP_m, \sum_{m=1}^{M} FN_m\right). \quad (16)$$

2) Macro – макроподход заключается в вычислении метрик для каждого из M классовых меток и взятия их среднего арифметического:

$$B_{\text{macro}} = \frac{1}{M}\sum_{m=1}^{M} B(TP_m, FP_m, TN_m, FN_m). \quad (17)$$

3) Weighted – взвешенный подход заключается в агрегации результатов классификации по каждому из M классовых меток отдельно по каждой метрике. После агрегации вычисляется *accuracy* для каждой классовой метки. Каждая метрика – *TP, FP, FN, TN* – нормируется на *accuracy* и вычисляется итоговая метрика:

$$B_{\text{Weighted}} = \\ = B\left(\sum_{m=1}^{M} \frac{TP_m}{A_m}, \sum_{m=1}^{M} \frac{TN_m}{A_m}, \sum_{m=1}^{M} \frac{FP_m}{A_m}, \sum_{m=1}^{M} \frac{FN_m}{A_m}\right). \quad (18)$$

Рабочий режим ИНС

Соотношение (12) представляет собой итерационную подачу элементов множества неразмеченных ЭД на вход нейронной сети. Результатом является упорядоченное множество меток $Labels_{\hat{n}}$, полученное в ходе итерационного прохождения строк $\hat{A}(\hat{n},\ )$ через ИНС: $Labels_{\hat{n}} = (set_{\hat{n}})$; $\hat{n} = \overline{1,\hat{N}}$.

Неразмеченные ЭД могут быть сформированы заранее или подаваться в потоковом режиме непосредственно с системных датчиков.

Отметим необходимость своевременной корректировки весов ИНС в рабочем режиме из-за влияния проблемы смены (дрейфа) концепта [23].

Многозначная классификация с помощью ИНС с множественным выходом

Эффективность функционирования ИНС связана, как правило, с удалением части связей между слоями (называемыми также *dropput* [24]), регуляризацией и нормировкой весовых коэффициентов на этапе обучения (15) [25], упрощением функций активации на уровне скрытых слоев [26].

Одной из главных проблем при использовании полносвязных ИНС в задачах многозначного анализа (классификация, прогнозирование) является неэффективное использование функции *softmax* в качестве функций выходного слоя. Функция *softmax* формирует нормированный выходной вектор, сумма элементов которого равна единице. При устремлении размерности классовых меток (совокупности состояний КС, ассоциированных с вторичными атрибутами и меток, ассоциированных с компьютерной атакой) в бесконечность, максимальное значение элемента такого нормированного вектора стремится к нулю. Данная проблема приводит к необходимости разработки дополнительного блока голосования, способного принимать решение о количестве классов, ассоциированных с таким вектором, превысивших некоторый порог принятия решения. При этом, поскольку количество классовых меток, потенциально ассоциированных с одной записью ЭД, ограничено только алфавитом, создание блока голосования является нетривиальной задачей. Без использования блока голосования (например, при попытке обойтись округлением в большую сторону), может

возникать проблема отказа от классификации для ряда записей в ЭД, как это иллюстрируется на рисунке 2.

Наиболее простым способом решения подобной проблемы принятия решений является ввод на последнем слое ИНС функции активации, не нормирующей выходной сигнал. В качестве такой функции может использоваться, например, сигмоидальная функция активации.

Поскольку решается задача многозначной классификации, обобщающая способность ИНС может быть повышена также за счет внедрения многозначности в сеть на структурном уровне. Под обобщающей способностью ИНС понимается эффективность решения задачи классификации на наборе данных, отсутствующем в обучающей выборке.

Модификация ИНС позволяет распараллеливать обработку информации по каждому целевому не только на уровне последнего слоя, но и на уровне скрытых слоев. Таким образом, начиная с последнего общего скрытого слоя, ИНС разбивается «на не связанные между собой ветви». «Ветви» могут интерпретироваться как отдельные полносвязные ИНС с отдельным выходом (рисунок 3). Подобные ИНС получили наименование «ИНС с множественным выходом» (многозначным выходом) [27, 28].

На этапе обучения ИНС с множественным выходом, при обратном распространении ошибки, градиент вычисляется независимо для каждой ветви вплоть до общих полносвязных слоев.

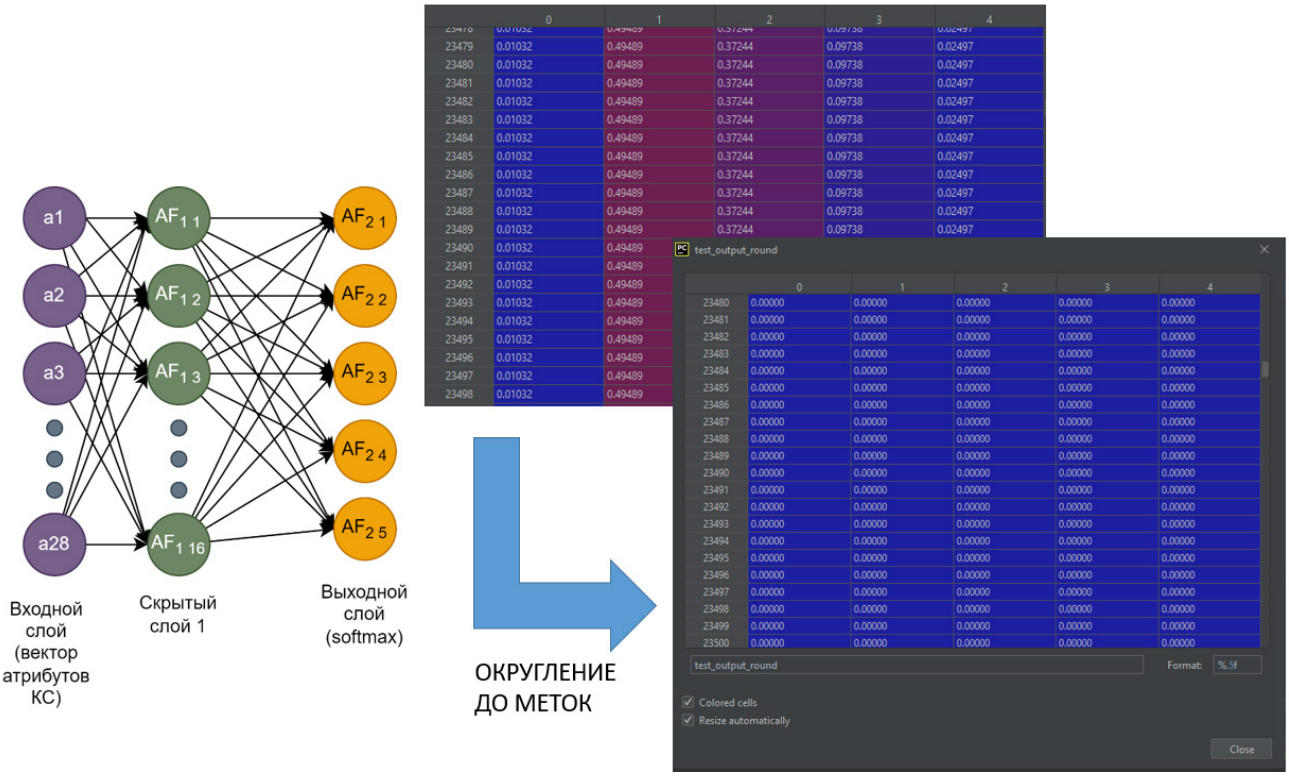

Рис. 2. Проблема принятия решений при использовании функции активации *softmax*

Fig. 2. Decision Problem When Using the Softmax Activation Function

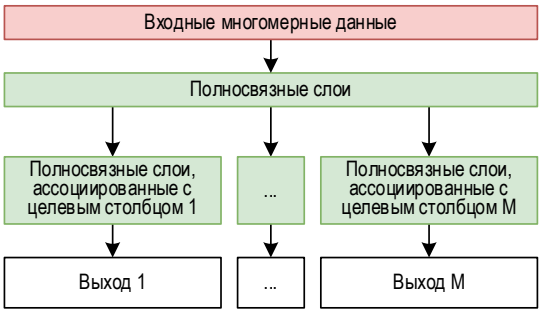

Рис. 3. Структура ИНС с множественным выходом

Fig. 3. Structure of ANN with Multiple Output

Иным подходом к повышению обобщающей способности может стать архитектура, аналогичная представленной на рисунке 3, но без общих полносвязных слоев. В результате каждая отдельная ИНС может быть представлена в виде бинарного классификатора по определенной классовой метке. Ансамбль таких бинарных классификаторов может быть сконфигурирован для решения задачи многозначной классификации (беггинговый метод классификации). Повышение качества классификации может быть достигнуто за счет установки модуля принятия решений, управляю-

щего обучением каждого бинарного классификатора (бустинговый метод классификации). Известны работы по созданию подобных бустинговых классификаторов из ансамбля полносвязных ИНС по принципу Random Forest [29].

Дополнительное повышение эффективности классификации может быть достигнуто за счет использования специальных подходов к обработке данных внутри ИНС таких как свертка, LSTM-блоки, блоки внимания и т. д.

Разработанная архитектура ИНС с множественным выходом базируется на структуре, изображенной на рисунке 3, и позволяет учитывать многозначность ЭД за счет использования общего полносвязного слоя.

Апробация разработанной архитектуры ИНС с множественным выходом на ЭД, содержащих компьютерную атаку

Оценка эффективности предложенного алгоритма и разработанной архитектуры ИНС с множественным выходом была проведена на примере ЭД, собранных в работах [30, 31]. Для обнаружения компьютерных атак с использованием ИНС в [32] разработан фреймворк *Kitsune*, а также предложен способ предобработки «сырых» ЭД в многомерную обезличенную табличную структуру с атрибутами метрического типа.

На рисунке 4 изображена топология КС IoT [32] для сбора данных, а также векторы, поясняющие происхождение атак. Захват сетевого трафика производился на маршрутизаторе в точках, указанных на рисунке цифрами. В каждом наборе данных первый миллион пакетов представлял собой чистый сетевой трафик, пакеты с номером миллион и выше содержали определенную компьютерную атаку.

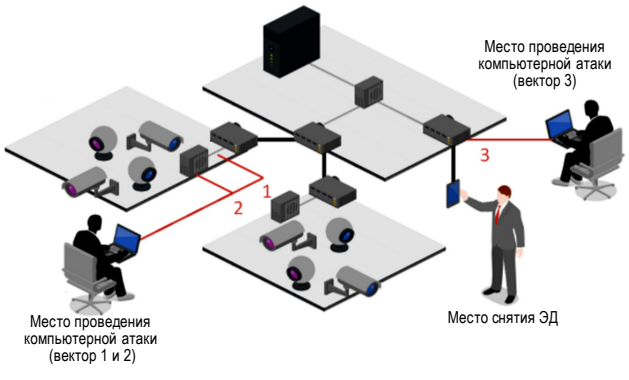

Рис. 4. Топология исследуемой сетевой инфраструктуры [32]

Fig. 4. Topology of the Analyzed Network Infrastructure [32]

При сборе информации с КС необработанные, «сырые» данные, поступающие с перечня устройств, захватывались в виде пакетов. Каждый пакет ассоциировался с временной меткой и рядом категориальных атрибутов таких как: MAC-адрес, IP-адрес, порты назначения и отправки и т. д. Преобразование пакетов в многомерные метрические векторы осуществлялось с использованием метода демпфированной инкрементной статистики, ДИС (DIS, *аббр. от англ.* Damped Incremental Statistics) [33].

ДИС ассоциировалось с параметром λ > 0, а также с тройкой $IS_\lambda = (\omega, LS, SS)$, где ω – число, линейная сумма и сумма квадратов пакетов, занесенных в инкрементную статистику. Каждая инкрементная статистика связана с потоками данных, определяемыми связкой MAC-адреса, IP-адресами, портами стека протоколов TCP/IP и четырьмя типами данных:

– IP отправителя (*srcIP*);

– MAC-адрес отправителя (*srcMAC*), включая пару ($srcMAC, srcIP$), ассоциированную с отправителем;

– информация, ассоциированная с каналом передачи данных – пара IP-адресов отправителя – получателя ($srcIP, dstIP$);

– сокет, ассоциированный с каналом передачи данных – в виде четверки: IP-адрес отправителя, порт отправителя, IP-адрес получателя, порт получателя ($srcIP, srcPort, dstIP, dstPort$).

Каждый новый пакет, поступающий на вход ДИС, обновлял статистику по правилам:

$$\gamma = 2^{-\lambda(t-t_{last})}; \quad \Delta IS_\lambda = (\gamma\omega + 1, \gamma LS + x, \gamma SS + x^2),$$

где t_{last} – отметка времени поступившего пакета, ассоциированного с потоком статистики; ΔIS_λ – приращение инкрементной статистики; параметр λ определяет интенсивность затухания статистики во времени.

Из инкрементной статистики был получен ряд одномерных и двумерных статистик, приведенных в таблице 1.

ТАБЛИЦА 1. Одномерные и двумерные статистики, выведенные с помощью ДИС

TABLE 1. Univariate and Bivariate Statistics Derived Using DIS

Размерность статистики	Наименование	Формула
Одномерная	Вес (ω)	ω
	Среднее (μ)	LS/ω
	Отклонение (σ)	$\sqrt{SS/\omega - \mu^2}$
Двумерная	Размер (O)	$\sqrt{\mu_i^2 - \mu_j^2}$
	Радиус (R)	$\sqrt{\sigma_i^4 - \sigma_j^4}$
	Ковариация ($Cov_{i,j}$)	$R_{i,j}/(\omega_i + \omega_j)$
	Коэффициент корреляции ($P_{i,j}$)	$Cov_{i,j}/(\sigma_i + \sigma_j)$

В [32] предложено использовать все статистики, представленные в таблице 1, для пяти значений λ: 5; 3; 1; 0,1; 0,01, что позволило сформировать после предобработки 115 атрибутов.

Поскольку наборы данных, сформированные для каждой из компьютерных атак, различны между собой по количеству пакетов, каждой атаке ставится в соответствие две .csv таблицы: таблица, ассоциированная с обезличенными ЭД (размерностью 115 атрибутов), и таблица, ассоциированная с целевым столбцом – бинарной классовой меткой о проведении (отсутствии) компьютерной атаки.

Размерности ЭД для каждого типа атаки существенно различаются. Наименьший объем данных ассоциировался с набором типа «*Mirai*» (компьютерная атака, направленная на заражения сети интернета вещей вредоносным программным обеспечением) – всего 750 тыс. записей. Наибольший объем был зафиксирован у атак типа «отказ в обслуживании» и «SSL Renegotiation» – более 6 млн пактов.

Для апробации разработанного алгоритма из всего множества наборов данных был выбран набор, соответствующий атаке «OS Scan», содержащий ~1,6 млн записей ЭД.

Предобработка ЭД для формирования многозначного набора классовых меток

Эксперимент подразумевал разделение входных данных на первичные и вторичные атрибуты КС. Поскольку истинное назначение каждого из 115 атрибутов ЭД неизвестно, их разделение производилось по следующему алгоритму.

Шаг 1. Весь набор данных по каждому атрибуту оценивался по совокупной информационной (индекс Джини, …) и статистической значимости атрибутов ЭД (важности атрибутов [6]) по отношению к целевому столбцу бинарных классовых меток, сигнализирующих о наличии (отсутствии) атаки типа «OS Scan».

Шаг 2. Внутри каждой группы полученных метрик производилась нормировка с целью получения оценок в диапазоне от 0 до 1. Группой считается группа метрик, вычисленная в рамках одного метода.

Шаг 3. Оценки, ассоциированные с каждой группой метрик, усреднялись.

Шаг 4. Атрибуты ЭД ранжировались по убыванию усредненной метрики оценки важности.

Шаг 5. Атрибуты ЭД, имевшие усредненную важность менее 0,1 – считались не влияющими на целевой столбец и выносились в множество вторичных атрибутов.

Шаг 6. Для вторичных атрибутов, полученных на шаге 5, по каждому атрибуту рассчитывалось среднеквадратическое отклонение.

Шаг 7. Для каждого вторичного атрибута формировался дополнительный целевой столбец классовых меток, полученных по правилу «если значение вторичного атрибута превышает среднеквадратичное отклонение – считать запись ЭД аномальной по данному атрибуту».

Шаг 8. Вторичные атрибуты исключались из ЭД с сохранением ассоциированных с ними классовых меток.

После выполнения приведенных восьми шагов ЭД содержат: первичные атрибуты, имеющие статистическую и информационную взаимосвязь с целевым столбцом, ассоциированным с компьютерной атакой «OS Scan»; целевой столбец, ассоциированный с компьютерной атакой «OS Scan; целевые столбцы, ассоциированные с вторичными атрибутами, не имеющие прямой (линейной) взаимосвязи ни с одним из первичных атрибутов ЭД.

Полученная таким образом каждая запись ЭД обладает множеством классовых меток, а сами записи являются многозначными.

Результат выполнения шагов 1–5 отражен на гистограмме (рисунок 5). По оси абсцисс отложены наименования атрибутов; по оси ординат – усредненные оценки важности соответствующих атрибутов. Пороговой линией отсечены атрибуты, не превышающие порог, введенный на шаге 5. Для удобства последующего анализа, каждый атрибут помечен идентификатором согласно шаблону «cXX», где XX – номер атрибута в ЭД Kitsune [32].

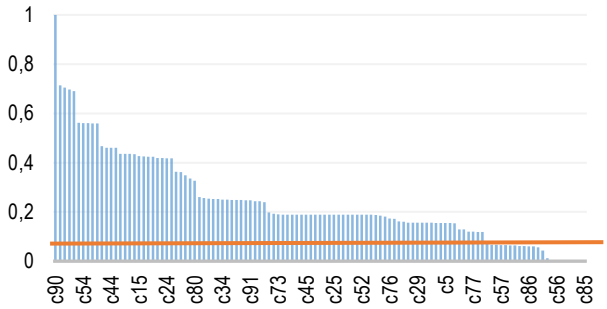

Рис. 5. Гистограмма распределения важности атрибутов ЭД с порогом отсечки

Fig. 5. Distribution Histogram of the Importance of Experimental Data Attributes With a Cutoff Threshold

Анализ гистограммы показывает, что при установленном пороге важности вторичными атрибутами являются 22. Их углубленный анализ показал, что не все атрибуты корректно использовать для порождения дополнительных классовых меток. По итогам исследования было сформировано 20 вторичных атрибутов ЭД и 95 первичных:

$$A_{\text{experiment}} = \left(A_{\text{втор }1}, A_{\text{втор }2}, \ldots, A_{\text{втор }20}\right) \cup \cup \left(A_{\text{перв }1}, A_{\text{перв }22}, \ldots, A_{\text{перв }95}\right).$$

Для каждого вторичного атрибута формировался дополнительный целевой столбец классовых меток, полученных по правилу: «если значение вторичного атрибута превышает среднеквадратичное отклонение – считать запись ЭД аномальной по данному атрибуту». Согласно шагу 8, вторичные атрибуты исключались из ЭД с сохранением ассоциированных с ними классовых меток.

С помощью библиотеки skikit-learn [34] входные ЭД были предобработаны (стандартизированы) путем приведения их к единичной дисперсии и нулевому среднему. Пропорция разделения ЭД была выбрана следующей: 45 % ассоциировано с обучающей выборкой; 22 % – с валидационной выборкой; 33 % – с тестовой выборкой. Поскольку часть классовых меток, включая метку об атаке *OS Scan,* встречается достаточно редко, ЭД дополнительно перемешивались. Отметим, что результирующее количество классовых меток обработанных ЭД, равно 21: 20 ассоциированы с вторичными атрибутами и 1 – с наличием (отсутствием) компьютерной атаки типа «*OS Scan*».

Детализация структуры, разработанной ИНС с множественным выходом

Архитектура разработанной ИНС с множественным выходом соответствует структуре, представленной на рисунке 3 с ограничениями, накладываемыми доступными вычислительными средствами. Поскольку вторичные атрибуты ЭД исключались из ЭД с сохранением ассоциированных с ними классовых меток, а результирующее количество классовых меток, обработанных ЭД, равно 21; мощность алфавита всех возможных классовых меток S (4) также равна 21:

$$S = \cup_{n=1}^{N} set_n = \left|\{s_{a_{90}}, s_{a_{83}}, \dots, s_{a_{85}}\} \cup \{s_{attack}\}\right| = 21.$$

Таким образом, для возможности анализа многозначного множества из 21 классовой метки необходимо сформировать ИНС с 21 выходом. Для учета взаимосвязи между метками необходимо задание общего слоя. Детализируем разработанную архитектуру. Для учета каждой «ветви» зададим структурную матрицу NN (8) в виде множества матрица $NN^{(tree)}$, где *(tree)* – идентификатор участка ИНС. В соответствии с условиями, всего число таких участков в данной задаче равно 22: участок, ассоциированный с общим слоем, и 21 участок, ассоциированный с ветвями ИНС.

Детализация разработанной ИНС имеет вид:

$$NN_{multioutput} = \left(NN^{(1)}, NN^{(2)}, \dots, NN^{(22)}\right).$$

Конфигурация ИНС формировалась с учетом особенностей, используемых в дальнейшем программных библиотек, имеющихся аппаратных вычислительных ресурсов и временных ограничений.

Итоговая конфигурация имеет вид:

$$Hyperparams^{multioutput} = (lossFunc, optimiser, epoch, batchSize),$$
$$optimiser = Adam;\ \ lossFunc = lossFunc_{BCE};\ epoch = 10;$$
$$batchSize = 512.$$

В качестве функции потерь для каждой ветви ИНС была выбрана бинарная перекрестная энтропия (*binary crossentropy*), а в качестве метрики оценки эффективности классификации на этапах обучения и валидации – метрика *accuracy*.

Выбор оптимальной эпохи обучения проводился с помощью разведочного обучения ИНС и анализа ландшафта функции потерь, а также соответствующей метрики. Диапазон изменения эпох для данной конфигурации был выбран в интервале 1–10.

Рассмотрим структуру первого слоя разработанной ИНС. Исследования показали, что наиболее эффективным по времени обучения и по выходным метрикам оценки качества классификации, с учетом метода обратного распространения ошибки, является задание функции активации типа *ReLU* с 64 нейронами:

$$NN^{(1)} = \begin{pmatrix} AF_1{}^{\mathrm{ReLU}}(net_1) \\ AF_2{}^{\mathrm{ReLU}}(net_2) \\ \dots \\ AF_{64}{}^{\mathrm{ReLU}}(net_{64}) \end{pmatrix},$$

где $AF_{lc}{}^{\mathrm{ReLU}}(net_{lc}) = \begin{cases} net_{lc}, net_{lc} > 0 \\ 0, net_{lc} <= 0 \end{cases}$.

Количество нейронов обусловлено вычислительными возможностями оборудования для проведения экспериментов и разумной достаточностью ширины слоя для большего числа ЭД [35].

Опишем *tree*-ю ветвь ИНС, $tree = \overline{2, N}$. Каждая ветвь состоит из семи слоев. Ветвь логически поделена на три зоны: первая и вторая подобны и состоят из тройки вида «*полносвязный слой с функцией активации ReLU – слой нормализации – слой dropout*».

Приведем запись первой тройки:

$$NN^{(tree)}(\ ,1) = \begin{pmatrix} AF_{1,1}{}^{\mathrm{ReLU}}(net_{1,1}) \\ AF_{2,1}{}^{\mathrm{ReLU}}(net_{2,1}) \\ \dots \\ AF_{32,1}{}^{\mathrm{ReLU}}(net_{32,1}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,2) = \begin{pmatrix} AF_{1,2}{}^{norm}(net_{1,2}) \\ AF_{2,2}{}^{norm}(net_{2,2}) \\ \dots \\ AF_{32,2}{}^{norm}(net_{32,2}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,3) = \begin{pmatrix} AF_{1,3}{}^{dropout}(net_{1,3}) \\ AF_{2,3}{}^{dropout}(net_{2,3}) \\ \dots \\ AF_{32,3}{}^{dropout}(net_{32,3}) \end{pmatrix}.$$

Для снижения влияния проблемы «затухающего градиента» [36], в качестве функции активации полносвязного слоя $NN^{(tree)}(\ ,1)$ задана функция типа *ReLU* с 32 нейронами. Слои нормализации и слои dropout выбраны одинаковой размерности, равной 32 нейронам.

Опишем вторую тройку слоев:

$$NN^{(tree)}(\ ,4) = \begin{pmatrix} AF_{1,4}{}^{\mathrm{ReLU}}(net_{1,4}) \\ AF_{2,4}{}^{\mathrm{ReLU}}(net_{2,4}) \\ \dots \\ AF_{16,4}{}^{\mathrm{ReLU}}(net_{16,4}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,5) = \begin{pmatrix} AF_{1,5}{}^{norm}(net_{1,5}) \\ AF_{2,5}{}^{norm}(net_{2,5}) \\ \dots \\ AF_{16,5}{}^{norm}(net_{16,5}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,6) = \begin{pmatrix} AF_{1,6}{}^{dropout}(net_{1,6}) \\ AF_{2,6}{}^{dropout}(net_{2,6}) \\ \dots \\ AF_{16,6}{}^{dropout}(net_{16,6}) \end{pmatrix}.$$

Как видно, количество нейронов, ассоциированных со второй тройкой, снижено до 16. Вторая тройка слоев необходима для «плавного» принятия решения присвоении *tree*-й метки класса. Проходя через слои размерностей 64, 32 и 16, входной вектор несколько раз преобразуется («кодируется»), до тех пор, пока не поступит на вход к последнему, седьмому, слою *tree*-й ветви.

Второй и пятый слои *tree*-й ветви ИНС состоят из слоя нормализации – специализированной функции, приводящей каждый элемент входного вектора к нулевому математическому ожиданию и единичной дисперсии.

Третий и шестые слои *tree*-й ветви ИНС состоят из функции «обнуления» части значений входного вектора. В результате часть связей между соседними слоями обрывается, что приводит к улучшению обобщающей способности ИНС и ускорению обучения. Как правило, количество удаляемых связей между нейронами устанавливается в виде доли в диапазоне от 0 (все связи остаются неизменными) до 1 (все связи между слоями обнуляются). Для разработанной архитектуры доля удаляемых связей для $NN^{(tree)}(\ ,3)$ выбрана 0,25 (25 % всех связей удаляется) и для $NN^{(tree)}(\ ,6)$ – выбрана равной 0,15.

Выход каждой ветви оканчивается полносвязным слоем, состоящим из одного нейрона с сигмоидальной функцией активации $NN^{(tree)}(\ ,7) = \left(AF_{1,7}{}^{sigm}(net_{1,7})\right)$.

Выбор сигмоидальной функции активации обусловлен постановкой задачи: каждая ветвь должна возвращать бинарную метку класса – 0 (метка, ассоциированная с ветвью – отсутствует) или 1 (метка, ассоциированная с ветвью – присутствует). Сигмоидальная функция активации подходит для решения задачи бинарной классификации (в контексте *tree*-й ветви ИНС).

Визуализируем разработанную архитектуру ИНС с множественным выходом (рисунок 6). Размерность входного атрибутного пространства ЭД соответствует $|A_{\text{experiment}}| = 95$; количество ветвей ИНС – количеству классовых меток $|S_{\text{experiomentt}}| = 21$.

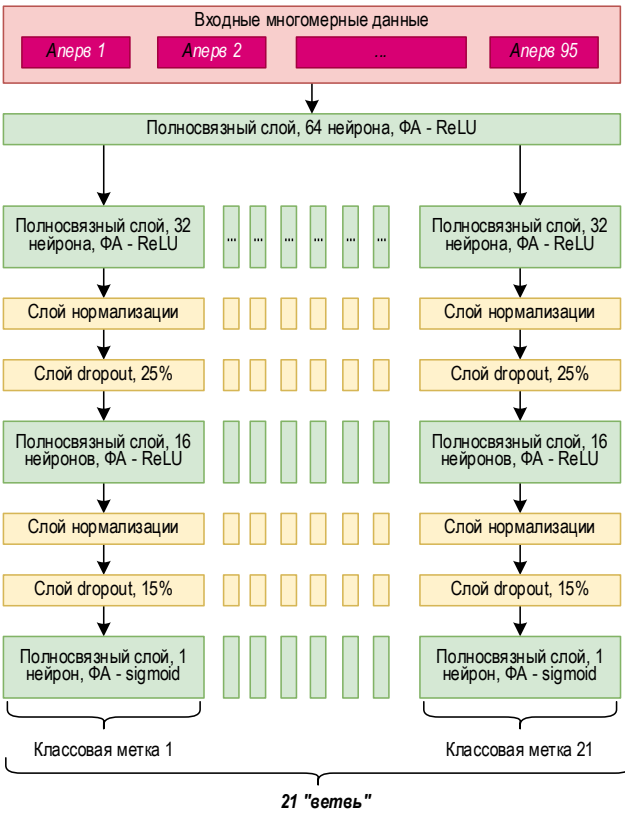

Рис. 6. Визуализация разработанной архитектуры ИНС с множественным выходом

Fig 6. Visualization of the Developed ANN Architecture with Multiple Outputs

Анализ результатов

Результаты работы разработанной архитектуры ИНС, полученные в результате эксперимента, оценивались на тестовой выборке по группе метрик оценки качества классификации, приведенных в таблице 2, для каждой из исследуемых эпох обучения (диапазон измерения – от 1 до 10).

Анализ полученных результатов позволяет судить о высокой скорости обучения разработанной ИНС. За 10 эпох метрика *accuracy* достигла значения ~0,999, как и остальные метрики оценки качества классификации. При этом наблюдается нелинейная зависимость эффективности классификации по приведенным метрикам.

ТАБЛИЦА 2. Результаты многозначной классификации по 21 классовой метке для ИНС с множественным выводом (тестовая выборка)

TABLE 2. Results of Multi-Label Classification by 21 Class Labels for ANNs with Multiple Output (Test Set)

Эпоха	*accuracy*	precision			recall			f1			ROC_AUC		
		micro	macro	weighted	micro	macro	weighted	micro	macro	weighted	OVO_micro	OVO_macro	OVO_weighted
1	0,9534	0,9937	0,9716	0,9960	0,9998	0,9996	0,9998	0,9968	0,9817	0,9975	0,9979	0,9984	0,9995
2	0,9589	0,9986	0,9750	0,9972	0,9962	0,9619	0,9962	0,9974	0,9655	0,9964	0,9976	0,9806	0,9979
3	0,9603	0,9998	0,9710	0,9970	0,9952	0,9523	0,9952	0,9975	0,9521	0,9952	0,9976	0,9761	0,9975
4	0,9597	0,9998	0,9705	0,9969	0,9952	0,9523	0,9952	0,9975	0,9522	0,9952	0,9975	0,9761	0,9975
5	0,9597	0,9999	0,9761	0,9976	0,9950	0,9521	0,9950	0,9975	0,9522	0,9952	0,9975	0,9760	0,9975
6	0,9603	0,9999	0,9729	0,9973	0,9952	0,9523	0,9952	0,9975	0,9523	0,9952	0,9975	0,9761	0,9975
7	0,9603	0,9998	0,9773	0,9976	0,9953	0,9524	0,9953	0,9975	0,9523	0,9952	0,9976	0,9761	0,9976
8	0,9608	0,9999	0,9911	0,9990	0,9953	0,9524	0,9953	0,9976	0,9525	0,9953	0,9976	0,9762	0,9976
9	0,9608	0,9999	0,9916	0,9991	0,9953	0,9524	0,9953	0,9976	0,9525	0,9953	0,9976	0,9762	0,9976
10	0,9990	0,9999	0,9999	0,9999	0,9999	0,9995	0,9999	0,9999	0,9997	0,9999	0,9999	0,9997	0,9999

Рассмотрим на примере метрик *accuracy*, $AUC_{OVO\ micro}$, $AUC_{OVO\ macro}$, $AUC_{OVO\ weighted}$ (рисунок 7).

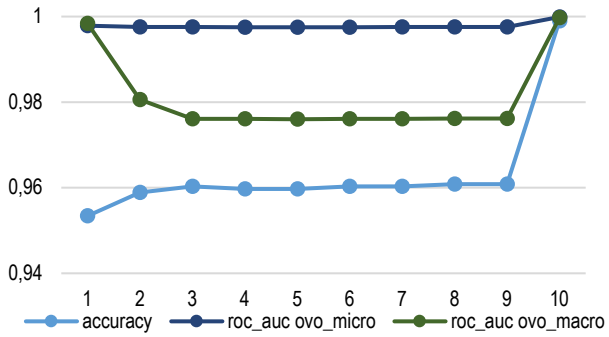

Рис. 7. Сравнение эффективности классификации по метрикам accuracy, $AUC_{OVO\ micro}$, $AUC_{OVO\ macro}$, $AUC_{OVO\ weighted}$ для 10 эпох обучения разработанной ИНС

Fig. 7. Comparison of Classification Efficiency by Metrics Accuracy, $AUC_{OVO\ micro}$, $AUC_{OVO\ macro}$, $AUC_{OVO\ weighted}$ for 10 Epochs of Training Developed by ANN

На промежутке от 1 до 9 эпохи наблюдалось плато эффективности классификации. Прирост по анализируемой метрике составил ~0,01. Вероятная причина столь малого прироста в эффективности классификации обусловлена корректировкой весовых коэффициентов ИНС в сторону минимума (локального или глобального) функции потерь. Наконец, на 10-й эпохе, веса ИНС скорректировались достаточно для корректной классификации почти всех записей ЭД, ассоциированных с тестовой выборкой.

Сравнительный анализ результатов многозначной классификации компьютерных атак разработанной архитектуры ИНС с множественным выходом

Сравним разработанную архитектуру ИНС с множественным выходом с известными аналогами и результатами, достигнутыми в [32]. Там же были рассмотрены известные алгоритмы обучения без учителя, базирующиеся на принципе обнаружения выбросов в многомерных данных: *Isolation Forests (IF)* и *Gaussian Mixture Models* (*GMM*), инкрементная реализация *GMM*, *pcStream2*.

Оценка эффективности обнаружения компьютерной атаки оценивалась по ряду метрик: *True Positive Rate* (*TPR*), *False Negative Rate* (*FNR*), AUC, *Equal Error Rate* (*EER*). Для сравнения результатов, полученных с помощью разработанной ИНС с множественным выходом с результатами работы [32], выбрана метрика AUC.

Сравнение результатов классификации, полученных с использованием разработанной ИНС с известными аналогами, осуществлялось для двух сценариев. В первом сценарии ЭД поступают на вход ИНС без разделения на первичные и вторичные атрибуты. Для обучения ИНС использовано исходное атрибутное пространство размерностью $M = 115$. В качестве целевого столбца использована только информация о наличии (отсутствии) атаки типа «*OS Scan*». Таким образом, выполнялась бинарная классификация.

Второй сценарий подразумевал предобработку ЭД согласно алгоритму:

$$A_{experiment} = \left(A_{втор\ 1}, A_{втор\ 2}, \ldots, A_{втор\ 20}\right) \cup \cup \left(A_{перв\ 1}, A_{перв\ 22}, \ldots, A_{перв\ 95}\right).$$

В случае второго сценария ЭД являлись многозначными, поскольку каждой записи соответствовала 21 классовая метка. Полученные результаты двух сценариев использования ИНС и данные известных аналогов по метрике AUC представлены в таблице 3. Первые 7 записей ассоциированы с известными решениями [32]: *Suricata, Iso. Forest, GMM*, Инкрементная реализация *GMM*, *pcStream, Kitsune*

(m = 10), *Kitsune* (m = 1). Позиции № 8 и 9 получены с помощью разработанной архитектурой ИНС с множественным выходом. Позиция № 8 связана с первым сценарием использования ИНС, соответствующем бинарной классификации (обнаружением) компьютерной атаки. Позиция № 9 соответствует результатам со сценарием многозначной классификации.

ТАБЛИЦА 3. Сравнительный анализ эффективности обнаружения компьютерной атаки типа «*OS Scan*», разработанной ИНС для разных алгоритмов

TABLE 3. Comparative Analysis of the Effectiveness of Detecting a Computer Attack Such as OS Scan Developed by ANN for Different Algorithms

№ п/п	Наименование алгоритма	AUC
1	Suricata	0,5000
2	Iso. Forest	0,9070
3	GMM	0,9493
4	Инкрементная реализация GMM	0,9469
5	pcStream	0,7419
6	Kitsune (m=10)	0,9481
7	Kitsune (m=1)	0,9481
8	ИНС с множественным выходом (только компьютерная атака)	0,9995
9	ИНС с множественным выходом (многозначный режим работы)	0,9999

Анализ данных, приведенных в таблице 3, показывает, что предложенная архитектура ИНС с множественным выходом доминирует над остальными алгоритмами, достигая наибольшей точности классификации. В случае бинарной классификации (атрибутное пространство размерностью M = 115) выигрыш составляет 5,2 % по сравнению с наибольшей достигнутой точностью алгоритма *GMM*. При многозначной классификации выигрыш составляет 5,4 %. В обоих случаях выигрыш может быть объяснен преимуществами предложенной архитектуры ИНС с множественным выходом.

Выигрыш многозначной реализации обуславливается наличием дополнительных «ветвей» в ИНС, и возникающих нелинейных взаимосвязей между первым (общим) слоем $NN^{(1)}$ и первой тройкой каждой *tree*-й ветви:

$$NN^{(tree)}(\ ,1) = \begin{pmatrix} AF_{1,1}{}^{\mathrm{ReLU}}(net_{1,1}) \\ AF_{2,1}{}^{\mathrm{ReLU}}(net_{2,1}) \\ \dots \\ AF_{32,1}{}^{\mathrm{ReLU}}(net_{32,1}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,2) = \begin{pmatrix} AF_{1,2}{}^{norm}(net_{1,2}) \\ AF_{2,2}{}^{norm}(net_{2,2}) \\ \dots \\ AF_{32,2}{}^{norm}(net_{32,2}) \end{pmatrix},$$

$$NN^{(tree)}(\ ,3) = \begin{pmatrix} AF_{1,3}{}^{dropout}(net_{1,3}) \\ AF_{2,3}{}^{dropout}(net_{2,3}) \\ \dots \\ AF_{32,3}{}^{dropout}(net_{32,3}) \end{pmatrix}.$$

Достоинствами предложенной архитектуры ИНС являются учет многозначных закономерностей между классовыми метками на этапе обучения за счет использования общего первого слоя, масштабируемость к любому числу классовых меток, высокая обобщающая способность ИНС и быстрая сходимость.

Среди недостатков предложенной архитектуры стоит отметить значительное потребление временных и вычислительных ресурсов на этапе обучения ИНС при большом количестве ветвей. Предложенная архитектура создана с использованием библиотеки *Keras*. Потенциальным решением проблемы высоких вычислительных затрат может стать смена библиотеки с *Keras* на *Pytorch*.

Заключение

Формализация ИНС в терминах матричной алгебры позволяет учитывать случай многозначной классификации. Достоинством предложенной формализации является лаконичность ряда записей, ассоциированных с рабочим режимом работы ИНС и режимом обучения.

Предложенный ансамблевый алгоритм оценки статистической и информационной значимости атрибутов ЭД в контексте одномерного целевого столбца позволяет обнаруживать как наиболее значимые атрибуты ЭД, так и ряд независимых от целевого столбца атрибутов.

Предложенная архитектура ИНС с множественным выходом позволяет решать задачи обнаружения и классификации многозначных компьютерных атак в среднем на 5 % эффективнее известных аналогов. Повышение эффективности обусловлено учетом многозначных закономерностей между классовыми метками на этапе обучения за счет использования общего первого слоя.

Достоинствами предложенной архитектуры ИНС является масштабируемость к любому числу классовых меток, высокая обобщающая способность ИНС и быстрая сходимость. В тоже время высокие затраты временных и вычислительных ресурсов можно устранить путем оптимизации программного кода или использования иных программных библиотек.

Список источников

1. Большаков А.С. Губанкова Е.В. Обнаружение аномалий в компьютерных сетях с использованием методов машинного обучения // REDS: Телекоммуникационные устройства и системы. 2020. Т. 10. № 1. С. 37–42.
2. Sheluhin O.I., Rakovskiy D.I. Multi-Label Learning in Computer Networks // Proceedings of the Conference at 2023 Systems of Signals Generating and Processing in the Field of on Board Communications (Moscow, Russian Federation, 14–16 March 2023). IEEE, 2023. DOI:10.1109/IEEECONF56737.2023.10092157
3. Чечулин А.А. Проблемы сбора корректной и непротиворечивой информации о состоянии компьютерной сети // Информатизация и связь. 2023. № 1. С. 91–94. DOI:10.34219/2078-8320-2023-14-1-91-94
4. Шелухин О.И., Раковский Д.И. Прогнозирование профиля функционирования компьютерной системы на основе многозначных закономерностей // Вопросы кибербезопасности. 2022. № 6(52). С. 53–70. DOI:10.21681/2311-3456-2022-6-53-70
5. Sheluhin O.I., Osin A.V., Rakovsky D.I. New Algorithm for Predicting the States of a Computer Network Using Multivalued Dependencies // Automatic Control and Computer Sciences. 2023. Vol. 57. Iss. 1. PP. 48–60. DOI:10.3103/S0146411623010091
6. Rakovskiy D.I. Analysis of the problem of multivalued of class labels on the security of computer networks» // Synchroinfo journal. 2022. Iss. 6. PP. 10–17. DOI:10.36724/2664-066X-2022-8-6-10-17
7. Васильев В.И., Вульфин А.М., Гвоздев В.Е., Картак В.М., Атарская Е.А. Обеспечение информационной безопасности киберфизических объектов на основе прогнозирования и обнаружения аномалий их состояния // Системы управления, связи и безопасности. 2021. № 6. С. 90–119. DOI:10.24412/2410-9916-2021-6-90-119
8. Sheluhin O.I., Barkov V.V., Sekretarev S.A. The online classification of the mobile applications traffic using data mining techniques // T-Comm. 2019. Т. 13. № 10. С. 60–67. DOI:10.24411/2072-8735-2018-10317
9. Шелухин О.И., Барков В.В., Полковников М.В. Классификация зашифрованного трафика мобильных приложений методом машинного обучения // Вопросы кибербезопасности. 2018. № 4(28). С. 21–28. DOI:10.21681/2311-3456-2018-4-21-28
10. Ismailov V.E. A three layer neural network can represent any multivariate function // Journal of Mathematical Analysis and Applications. 2023. Vol. 523. Iss. 1. P. 127096. DOI:10.1016/j.jmaa.2023.127096
11. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., et al. Attention Is All You Need // arXiv:1706.03762v7. 2017. DOI:10.48550/arXiv.1706.03762
12. Elbayad M., Besacier L., Verbeek J. Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction // arXiv:1808.03867v3. 2018. DOI:10.48550/arXiv.1808.03867
13. Евграфов В.А., Ильюшин Е.А. Спайковые нейронные сети // International Journal of Open Information Technologies. 2021. Т. 9. № 7. С. 21–31.
14. Trentin E. Multivariate Density Estimation with Deep Neural Mixture Models // Neural Processing Letters. 2023. Vol. 53. Iss. 2. PP. 1–17. DOI:10.1007/s11063-023-11196-2
15. Воронцов К.В. Математические методы обучения по прецедентам (теория обучения машин). URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf (дата обращения 17.05.2023)
16. Молодцов Д.А. Сравнение и продолжение многозначных зависимостей // Нечеткие системы и мягкие вычисления. 2016. Т. 11. № 2. С. 115–145.
17. Olson D.L., Araz Ö.M. Cluster Analysis // Data Mining and Analytics in Healthcare Management. International Series in Operations Research & Management Science. Cham: Springer, 2023. Vol. 341. PP. 53–68. DOI:10.1007/978-3-031-28113-6_5
18. Молодцов Д.А., Осин А.В. Новый метод применения многозначных закономерностей // Нечеткие системы и мягкие вычисления. 2020. Т. 15. № 2. С. 83–95. DOI: 10.26456/fssc72
19. Кафтанников И.Л., Парасич А.В. Проблемы формирования обучающей выборки в задачах машинного обучения // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2016. Т. 16. № 3. С. 15–24. DOI:10.14529/ctcr160302
20. Javed R.K., Ayub N., Shiraz M. A Novel Approach Using Deep Learning for Network Based Intrusion Detection System // Thesis for: MS CS Advisor: Nasir Ayub and Prof. Dr. Muhammad Shiraz. DOI:10.13140/RG.2.2.21108.01922
21. Camargo J.T.F., Veraszto E.V., Barreto G., Amaral S.F Neural Networks and the Study of Time Series: An Application in Engineering Education // Journal of Mechanics Engineering and Automation. 2015. Vol. 5. P. 2159-5275153-160. DOI:10.17265/2159-5275/2015.03.003
22. Andrychowicz M., Denil M., Gómez S., Hoffman M., Pfau D., Schaul T., et al. Learning to learn by gradient descent by gradient descent // arXiv:1606.04474v2. 2016. DOI:10.48550/arXiv.1606.04474
23. Sheluhin O.I., Rybakov S.Y., Vanyushina A.V. Detection of Network Anomalies with the Method of Fixing Jumps of the Fractal Dimension in the Online Mode // Wave Electronics and Its Application in Information and Telecommunication Systems. 2022. Vol. 5. Iss. 1. PP. 430–435.
24. Kox J.H.A.M., van der Zwan J.S, Groenewoud J.H., Runhaar J., Bierma-Zeinstra S.M.A., Bakker E.J.M., et al. Predicting late dropout from nursing education or early dropout from the profession // Science Talks. 2022. Vol. 5. P. 100106. DOI:10.1016/j.sctalk.2022.100106
25. Lamia A.N.M. Role of data normalization in k-means algorithm results // Al-Kadhum 2nd International Conference on Modern Applications of Information and Communication Technology (Baghdad, Iraq, 8–9 December 2021). 2023. DOI:10.1063/5.0119267
26. Avant T., Morgansen K.A. Analytical Bounds on the Local Lipschitz Constants of ReLU Networks // IEEE Transactions on Neural Networks and Learning Systems. 2023. PP. 1–12. DOI:10.1109/TNNLS.2023.3273228
27. Bressan R. Building a multi-output Convolutional Neural Network with Keras // Medium. URL: https://towardsdatascience.com/building-a-multi-output-convolutional-neural-network-with-keras-ed24c7bc1178 (дата обращения 28.06.2023)

28. Do N.-T., Hoang V.-P., Doan V.S. A novel non-profiled side channel attack based on multi-output regression neural network // Journal of Cryptographic Engineering. 2023. DOI:10.1007/s13389-023-00314-4

29. Prasad J.R., Saikumar S., Subbarao B.V. Design and Development of Financial Fraud Detection using Machine Learning // International Journal of Emerging Trends in Engineering Research. 2020. Vol. 8. Iss. 9. PP. 5838–5843. DOI:10.30534/ijeter/2020/152892020

30. Kitsune Network Attack Dataset // Kaggle. URL: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune (дата обращения 22.02.2023)

31. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection // Github. URL: https://github.com/ymirsky/Kitsune-py (дата обращения 22.02.2023)

32. Mirsky Y., Doitshman T., Elovici Y., Shabtai A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection // arXiv:1802.09089. 2018. URL: https://arxiv.org/pdf/1802.09089.pdf (дата обращения 28.08.2023)

33. Miyamoto K., Goto H., Ishibashi R., Han C., Ban T., Takahashi T. et al. Malicious Packet Classification Based on Neural Network Using Kitsune Features // Poceedings of the Second International Conference on Intelligent Systems and Pattern Recognition (ISPR, Hammamet, Tunisia, 24–26 March 2022). Communications in Computer and Information Science. Cham: Springer, 2022. Vol. 1589. PP. 306–314. DOI:10.1007/978-3-031-08277-1_25

34. Preprocessing data // Scikit-Learn. URL: https://scikit-learn.org/stable/modules/preprocessing.html (дата обращения 28.06.2023)

35. Лукьянова О.А., Никитин О.Ю., Кунин А.С. Применение матричных фильтров и теории кос для процедурной генерации архитектур нейронных сетей // Вычислительные технологии. 2019. Т. 24. № 6. С. 69–78. DOI:10.25743/ICT.2019.24.6.009

36. Scheliga D., Maeder P., Seeland M. Dropout Is NOT All You Need to Prevent Gradient Leakage // Proceedings of the 37th AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence (Washington, USA, 7–14 February 2023). AAAI Press, 2023. Vol. 37. № 8. PP. 9733–9741. DOI:10.1609/aaai.v37i8.26163

References

1. Bol'shakov A.S. Gubankova E.V. Anomaly detection in computer networks using machine learning methods. *REDS*. 2020;10(1):37–42.

2. Sheluhin O.I., Rakovskiy D.I. Multi-Label Learning in Computer Networks. *Proceedings of the conference at 2023 Systems of Signals Generating and Processing in the Field of on Board Communications, 14–16 March 2023, Moscow, Russian Federation*. IEEE; 2023. DOI:10.1109/IEEECONF56737.2023.10092157

3. Chechulin A.A. The issues of collecting correct and consistent information about a computer network. *Informatization and communication*. 2023;1:91–94. DOI:10.34219/2078-8320-2023-14-1-91-94

4. Shelukhin O., Rakovskiy D. Prediction of the profile functioning of a computer system based on multivalued patterns. *Voprosy kiberbezopasnosti*. 2022;6(52):53–70. DOI:10.21681/2311-3456-2022-6-53-70

5. Sheluhin O.I., Osin A.V., Rakovsky D.I. New Algorithm for Predicting the States of a Computer Network Using Multivalued Dependencies. *Automatic Control and Computer Sciences*. 2023;57(1):48–60. DOI:10.3103/S0146411623010091

6. Rakovskiy D.I. Analysis of the problem of multivalued of class labels on the security of computer networks». *Synchroinfo journal*. 2022;6:10–17. DOI:10.36724/2664-066X-2022-8-6-10-17

7. Vasilyev V.I., Vulfin A.M., Gvozdev V.E., Kartak V.M., Atarskaya E.A. Ensuring information security of cyber-physical objectsbased on predicting and detecting anomalies in their state. *Systems of Control, Communication and Security*. 2021;6:90–119. DOI:10.24412/2410-9916-2021-6-90-119

8. Sheluhin O.I., Barkov V.V., Sekretarev S.A. The online classification of the mobile applications traffic using data mining techniques. *T-Comm*. 2019;13(10):60–67. DOI:10.24411/2072-8735-2018-10317

9. Shelukhin O., Barkov V., Polkovnikov M. Classification of encrypted mobile app traffic using the machine learning method. *Voprosy kiberbezopasnosti*. 2018;4(28):21–28. DOI:10.21681/2311-3456-2018-4-21-28

10. Ismailov V.E. A three layer neural network can represent any multivariate function. *Journal of Mathematical Analysis and Applications*. 2023;523(1):127096. DOI:10.1016/j.jmaa.2023.127096

11. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., et al. Attention Is All You Need. *arXiv:1706.03762v7*. 2017. DOI:10.48550/arXiv.1706.03762

12. Elbayad M., Besacier L., Verbeek J. Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction. *arXiv:1808.03867v3*. 2018. DOI:10.48550/arXiv.1808.03867

13. Evgrafov V., Il'yushin E. On spiking neural networks. *International Journal of Open Information Technologies*. 2021;9(7):21–31.

14. Trentin E. Multivariate Density Estimation with Deep Neural Mixture Models. *Neural Processing Letters*. 2023;53(2):1–17. DOI:10.1007/s11063-023-11196-2

15. Vorontsov K.V. *Mathematical methods of learning by precedents (machine learning theory)*. URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf [Accessed 17.05.2023]

16. Molodtsov D.A. Comparison and continuation of multivalued dependencies. *Fuzzy systems and soft computing*. 2016;11(2):115–145.

17. Olson D.L., Araz Ö.M. Cluster Analysis. *In: Data Mining and Analytics in Healthcare Management. International Series in Operations Research & Management Science, vol.341*. Cham: Springer; 2023. p.53–68. DOI:10.1007/978-3-031-28113-6_5

18. Molodtsov D.A., Osin A.V. A new method of applying multi-valued dependencies. *Fuzzy systems and soft computing*. 2020;15(2):83–95. DOI:10.26456/fssc72

19. Kaftannikov I.L., Parasich A.V. Problems of Training Set's Formation in Machine Learning Tasks. *Bulletin of the South Ural State University. Series "Computer Technologies, Automatic Control, Radio Electronics"*. 2016;16(3):15–24. DOI:10.14529/ctcr160302

20. Javed R.K., Ayub N., Shiraz M. A Novel Approach Using Deep Learning for Network Based Intrusion Detection System. *Thesis for: MS CS Advisor: Nasir Ayub and Prof. Dr. Muhammad Shiraz*. DOI:10.13140/RG.2.2.21108.01922

21. Camargo J.T.F., Veraszto E.V., Barreto G., Amaral S.F Neural Networks and the Study of Time Series: An Application in Engineering Education. *Journal of Mechanics Engineering and Automation*. 2015;5:2159-5275153-160. DOI:10.17265/2159-5275/2015.03.003

22. Andrychowicz M., Denil M., Gómez S., Hoffman M., Pfau D., Schaul T., et al. Learning to learn by gradient descent by gradient descent. *arXiv:1606.04474v2*. 2016. DOI:10.48550/arXiv.1606.04474

23. Sheluhin O.I., Rybakov S.Y., Vanyushina A.V. Detection of Network Anomalies with the Method of Fixing Jumps of the Fractal Dimension in the Online Mode. *Wave Electronics and Its Application in Information and Telecommunication Systems*. 2022;5(1):430–435.

24. Kox J.H.A.M., van der Zwan J.S, Groenewoud J.H., Runhaar J., Bierma-Zeinstra S.M.A., Bakker E.J.M., et al. Predicting late dropout from nursing education or early dropout from the profession. *Science Talks*. 2022. Vol. 5. P. 100106. DOI:10.1016/j.sctalk.2022.100106

25. Lamia A.N.M. Role of data normalization in k-means algorithm results. *Al-Kadhum 2nd International Conference on Modern Applications of Information and Communication Technology, 8–9 December 2021, Baghdad, Iraq*. 2023. DOI:10.1063/5.0119267

26. Avant T., Morgansen K.A. Analytical Bounds on the Local Lipschitz Constants of ReLU Networks. *IEEE Transactions on Neural Networks and Learning Systems*. 2023:1–12. DOI:10.1109/TNNLS.2023.3273228

27. Bressan R. Building a multi-output Convolutional Neural Network with Keras. *Medium*. URL: https://towardsdatascience.com/building-a-multi-output-convolutional-neural-network-with-keras-ed24c7bc1178 [Accessed 28.06.2023]

28. Do N.-T., Hoang V.-P., Doan V.S. A novel non-profiled side channel attack based on multi-output regression neural network. *Journal of Cryptographic Engineering.* 2023. DOI:10.1007/s13389-023-00314-4

29. Prasad J.R., Saikumar S., Subbarao B.V. Design and Development of Financial Fraud Detection using Machine Learning. *International Journal of Emerging Trends in Engineering Research*. 2020;8(9):5838–5843. DOI:10.30534/ijeter/2020/152892020

30. *Kaggle.* Kitsune Network Attack Dataset // kaggle. URL: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune [Accessed 22.02.2023]

31. *Github.* Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. URL: https://github.com/ymirsky/Kitsune-py [Accessed 22.02.2023]

32. Mirsky Y., Doitshman T., Elovici Y., Shabtai A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *arXiv:1802.09089*. 2018. URL: https://arxiv.org/pdf/1802.09089.pdf [Accessed 28.08.2023]

33. Miyamoto K., Goto H., Ishibashi R., Han C., Ban T., Takahashi T. et al. Malicious Packet Classification Based on Neural Network Using Kitsune Features. *Proceedings of the Second International Conference on Intelligent Systems and Pattern Recognition (ISPR, Hammamet, Tunisia, 24–26 March 2022). Communications in Computer and Information Science, vol.1589*. Cham: Springer; 2022. p.306–314. DOI:10.1007/978-3-031-08277-1_25

34. *Scikit-Learn*. Preprocessing data. URL: https://scikit-learn.org/stable/modules/preprocessing.html [Accessed 28.06.2023]

35. Lukyanova O.A., Nikitin O.Yu., Kunin A.S. Application of matrix filters and braid theory for the procedural generation of neural network architectures. *Computional Technologies*. 2019;24(6):69–78. DOI:10.25743/ICT.2019.24.6.009

36. Scheliga D., Maeder P., Seeland M. Dropout Is NOT All You Need to Prevent Gradient Leakage. *Proceedings of the 37th AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, 7–14 February 2023, Washington, USA, vol.37. №8*. AAAI Press; 2023. p.9733–9741. DOI:10.1609/aaai.v37i8.26163

Статья поступила в редакцию 04.07.2023; одобрена после рецензирования 18.07.2023; принята к публикации 27.08.2023.
The article was submitted 04.07.2023; approved after reviewing 18.07.2023; accepted for publication 27.08.2023.

Информация об авторах:

ШЕЛУХИН Олег Иванович — доктор технических наук, профессор, заведующий кафедрой «Информационная безопасность» Московского технического университета связи и информатики
https://orcid.org/0000-0001-7564-6744

РАКОВСКИЙ Дмитрий Игоревич — аспирант кафедры «Информационная безопасность» Московского технического университета связи и информатики
https://orcid.org/0000-0001-7689-4678