ИССЛЕДОВАНИЕ АРХИТЕКТУР ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ СО СВЕРТОЧНЫМИ И РЕКУРРЕНТНЫМИ СЛОЯМИ ДЛЯ ЗАДАЧ РАСПОЗНАВАНИЯ АНОМАЛИЙ СЕТЕВОГО ТРАФИКА В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Амосов О.С., Амосова С.Г.,
Институт проблем управления им. В.А. Трапезникова РАН
osa18@yandex.ru, amosovasg@yandex.ru
Иванов Ю.С., Жиганов С.В.
Комсомольский-на-Амуре государственный университет
ivanov_ys@icloud.com, zhiganov@knastu.ru

Аннотация. Предлагается распознавание аномалий сетевого трафика с использованием различных архитектур глубоких нейронных сетей. Проведен эксперимент по обнаружению DoS атак. Выявлены влияния комбинаций слоев нейронных сетей на характеристики точности. Новым является усиление классификации путем подкрепления входного вектора его кластерной оценкой.

Ключевые слова: аномалия сетевого трафика, сетевая атака, классификация, глубокая нейронная сеть, DoS-атака.

Введение

Распознавание аномального трафика, вызванного сетевой атакой, является одной из актуальных проблем защиты информации в корпоративных сетях. Наиболее распространенной атакой является DoS-атака (англ. Denial of Service) или ее вариант, распределенный по нескольким атакующим узлам – DDoS-атака (англ. Distributed Denial of Service), приводящие к отказу в обслуживании.

Сетевые системы обнаружения вторжений (ССОВ, англ. Network-based Intrusion Detection System, NIDS) анализируют сетевой трафик на наличие шаблонов в пакетах данных – «сигнатур» для обнаружения нарушений безопасности. Используя базу данных хорошо известных типов вторжений, ССОВ (NIDS) на основе «сигнатур» может быстро идентифицировать вторжения и инициировать соответствующий курс действий.

Для решения данной задачи авторами [1] предлагается модель системы обнаружения вторжений с использованием решающих деревьев.

В работах [2, 3] предложено идентифицировать сетевые угрозы с помощью фрактального и вейвлет-анализа. Авторы [4] предложили адаптивный подход на основе генетического алгоритма, что позволило им добиться точности распознавания аномального трафика 92,85%.

Перспективным также представляется использование аппарата нечеткой логики. Авторы [5] применили механизм интерполяции нечетких правил для обнаружения вторжений в протоколе MQTT.

В большинстве случаев для обучения моделей применяются общедоступные наборы данных (датасеты). Интерес представляет набор данных CICIDS2017 [6], содержащий самые распространенные современные атаки в виде временных рядов в формате библиотеки (англ. Packet Capture). Авторы представили результаты распознавания атак различными методами машинного обучения.

В области информационной безопасности нашли применение и глубокие нейронные сети. В работе [7] обнаруживают вторжения с точностью 91% с использованием глубокого многослойного персептрона (DMLP). Авторы предложили распознавать различные виды атак следующими глубокими моделями нейронных сетей (НС): рекуррентной нейронной сетью (англ. Recurrent neural network, RNN), сложенной рекуррентной нейронной сетью (англ. Stacked RNN) и сверточной нейронной сетью (англ. convolutional neural network, CNN). Для классификации сетевых атак рассмотрены также структуры глубоких нейронных сетей на основе 1D-сверточных и рекуррентных LSTM (англ. Long Short-Term Memory) слоев [8].

В настоящее время в связи с большим интересом к глубоким нейронным сетям (ГНС) представляет интерес разработка эффективного по точности и быстродействию метода интеллектуального анализа аномалий сетевого трафика в режиме реального времени на основе глубоких НС. Этому и посвящена статья.

1 Постановка задачи

Под сетевым трафиком будем понимать исходящую или входящую информацию из внешней сети, измеряемую в пакетах за единицу времени. Сетевой пакет – это определенным образом оформленный блок данных, который состоит из служебной информации, включающей стартовые биты (англ. preamble), заголовки (англ. headers), прицеп (англ. trailer), и полезной нагрузки (англ. payload).

Под аномальным трафиком будем понимать последовательность сетевых пакетов, вызванную вирусной сетевой активностью, действиями злоумышленников или неисправностью оборудования.

Тогда по входящему сетевому трафику необходимо обнаружить образ сетевого трафика и, выделив ключевые признаки, отнести его к одному из классов, после чего принять решение об отсутствии или наличии сетевой атаки для ее отражения.

Пусть имеются: множество образов сетевого трафика $\omega \in \Omega$, заданных признаками x_i, $i = \overline{1,n}$, совокупность которых для образа ω представлена векторными описаниями $\mathbf{\Phi}(\omega) = (x_1(\omega), x_2(\omega), ..., x_n(\omega)) = \mathbf{x}$; множество классов $\mathrm{B} = \{\beta_1, ..., \beta_k, ..., \beta_c, \}$, c – количество классов. Априорная информация представлена обучающим множеством (датасетом) $\mathrm{D} = \left\{ \left(\mathbf{x}^j, \beta^j \right) \right\}, j = \overline{1,L}$, заданным таблицей, каждая строка j которой содержит векторное описание образа $\mathbf{\Phi}(\omega)$ и метку класса β_k, $k = \overline{1,c}$. Заметим, что обучающее множество характеризует неизвестное отображение $^{*}\mathbf{F}: \Omega \to \mathrm{B}$.

Требуется по имеющимся пакетам $\mathbf{P}_t$ непрерывного сетевого трафика $\mathbf{N} = (\mathbf{P}_1, .., \mathbf{P}_t, .., \mathbf{P}_\tau)$ и априорной информации, заданной обучающим множеством $\mathrm{D} = \left\{ \left(\mathbf{x}^j, \beta^j \right) \right\}, j = \overline{1,L}$ для глубокого обучения НС с учителем, решить задачу распознавания образов: обнаружить образы ω в виде оценки признаков $\tilde{\mathbf{x}}$ с помощью отображения [9] $\mathbf{F}_1 : \mathbf{P}_t \to \tilde{\mathbf{x}}$ и классифицировать их с использованием отображения $\mathbf{F}_2 : \tilde{\mathbf{x}} \to \beta_k$, $k = \overline{1,c}$ в соответствии с заданным критерием $\mathrm{P}(\tilde{\mathbf{x}})$, минимизирующим вероятность ошибки.

Таким образом, необходимо найти отображение $\mathbf{F} : \mathbf{P}_t \to \beta_k$, $k = \overline{1,c}$, при котором $\mathbf{F}$ – является набором функций и алгоритмов f_i, $i = \overline{1, N_f}$.

Отображение $\mathbf{F} : \mathbf{P}_t \to \beta_k$ реализуется на основе предлагаемого вычислительного метода, содержащего следующие этапы: 1) предобработка данных; 2) уточнение области интереса; 3) выделение информативных признаков и классификация.

2 Решение задачи распознавания аномалий сетевого трафика

Стоит учесть, что в обучающем множестве, как правило, представлены отдельные сетевые пакеты $\mathbf{P}_t$, каждый из которых имеет метку класса. При этом необходимо выполнить классификацию последовательности пакетов, захваченных сканирующим окном $\mathbf{W}_t$ размером n и шагом смещения d. Исходные данные («сырые» пакеты), получаемые по сети, требуют предварительной обработки.

2.1 Предобработка данных

Для обучения и тестирования использовался датасет CICIDS2017 [6], содержащий следующие виды атак: Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet и DDoS.

Для увеличения количества обучающих примеров под DoS атакой будем понимать все виды DoS атак (DoS GoldenEye, DoS Hulk, DoS Slowhttp, DoS slowloris), представленных в датасете, включая DDoS.

Всего в датасете присутствуют данные, собранные за 5 дней – с понедельника по пятницу, в каждый из которых происходили различные атаки. Нами использовались данные о DoS атаках за 2 дня: среда (весь день) для обучения, пятница (после обеда) для тестирования.

Структура исходного «сырого» пакета представлена 84 признаками (рисунок 1), однако в формате для машинного обучения из структуры пакета исключены поля: идентификатор потока, IP адрес отправителя, порт отправителя, IP адрес получателя, порт получателя, тип протокола и временная метка. Итоговый вектор для обучения содержит 78 признаков.

Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets
Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	Bwd Packet Length Min
Bwd Packet Length Mean	Bwd Packet Length Std	Flow Bytes/s	Flow Packets/s	Flow IAT Mean	Flow IAT Std	Flow IAT Max	Flow IAT Min	Fwd IAT Total
Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max	Fwd IAT Min	Bwd IAT Total	Bwd IAT Mean	Bwd IAT Std	Bwd IAT Max	Bwd IAT Min
Fwd PSH Flags	Bwd PSH Flags	Fwd URG Flags	Bwd URG Flags	Fwd Header Length	Bwd Header Length	Fwd Packets/s	Bwd Packets/s	Min Packet Length
Max Packet Length	Packet Length Mean	Packet Length Std	Packet Length Variance	FIN Flag Count	SYN Flag Count	RST Flag Count	PSH Flag Count	ACK Flag Count
URG Flag Count	CWE Flag Count	ECE Flag Count	Down/Up Ratio	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Fwd Avg Bytes/Bulk
Fwd Avg Packets/Bulk	Fwd Avg Bulk Rate	Bwd Avg Bytes/Bulk	Bwd Avg Packets/Bulk	Bwd Avg Bulk Rate	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes
Init_Win bytes_forward	Init_Win bytes_backward	Act data_pkt_fwd	min_seg size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean
Idle Std	Idle Max	Idle Min						

Рис. 1. Структура исходного «сырого» пакета

Авторы датасета CICIDS2017 [6] представили результаты многоклассовой классификации собранных пакетов алгоритмами машинного обучения: KNN, RF, ID3, Adaboost, MLP, Naive-Bayes QDA. Авторами выделены следующие признаки как существенные при обнаружении DoS атак: стандартное отклонение длины пакетов в байтах, идущих в обратном направлении (B.Packet Len Std), протяженность потока по времени (Flow Duration), минимальное значение времени доставки потока в обоих направлениях (Flow IAT Min), среднее значение времени доставки потока в обоих направлениях (Flow IAT Mean), минимальное значение времени доставки потока в прямом направлении (Fwd IAT Min), средний размер каждого пакета (Avg Packet Size).

Авторы смогли достичь точности классификации атак 98%, однако ими проводилась классификация отдельных пакетов, тогда как в задаче обнаружения аномального трафика требуется обработка временного ряда. Стоит учесть, что применяемые методы являются стохастическими и важность параметров может меняться при каждом запуске модели. Предопределение отдельных признаков как более важных при обучении может привести к ошибкам. Поэтому необходимо провести нормализацию данных.

Нормализация признаков

Значения параметров в векторе для обучения имеют очень большой разброс (распределение) и масштаб, вследствие чего алгоритм машинного обучения может «решить», что один из признаков важнее другого, только исходя из их значений. Для устранения данного эффекта необходимо выполнить нормализацию при помощи метода StandardScaler (Z-score normalization), который приводит значение каждого признака к диапазону от -1,0 до 1,0, предотвращая тем самым неправильное поведение классификатора. Стандартная оценка для признака x_i рассчитывается как:

$$(1) \quad z_i = \frac{x_i - \mu}{\sigma},\ \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i),\ \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2},$$

где μ – среднее значение обучающих образцов или ноль, σ – стандартное отклонение обучающих образцов или единица, N – количество примеров.

Центрирование и масштабирование выполняется независимо для каждого признака по всей выборке. Значения μ и σ сохраняются для преобразования новых данных. В результате нормализации каждый пакет представлен вектором $\mathbf{z} = (z_1, z_2, \ldots, z_{78})$, размером 1×78, причем значение каждого элемента z_i находится в диапазоне $[-1 \ldots 1]$.

Высокая корреляция признаков приводит к снижению производительности обобщения данных из-за высокой дисперсии и меньшей интерпретируемости модели. Для выявления взаимосвязи между признаками, в машинном обучении используется коэффициент корреляции Пирсона. Это мера интенсивности и направления линейной зависимости между двумя переменными. Значение +1 означает идеально линейную положительную зависимость, а -1 означает идеально линейную отрицательную зависимость.

Для визуализации анализа корреляции по всем признакам обучающего множества была построена тепловая карта (Рисунок 2). Очевидно, что некоторые признаки в сетевом пакете

обладают высокой корреляцией (темные на рисунке), а значит, обучающий набор требует предварительной обработки в виде удаления признаков или кластеризации.

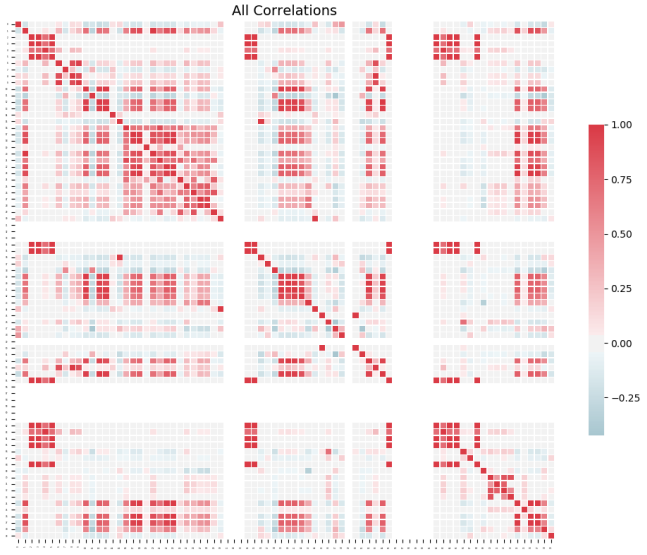

Рис. 2. Тепловая карта корреляции признаков

Кластеризация обучающей выборки

Для снижения размерности, обобщения и выявления скрытых закономерностей необходимо провести кластеризацию обучающей выборки. Кластеризация проводилась на обучающей части датасета, в которой присутствовало 97 686 примеров DoS атак и 128 025 нормальных пакетов.

В качестве алгоритма кластеризации предлагается использовать двуэтапный алгоритм BIRCH [10] (англ. Balanced Iterative Reducing and Clustering using Hierarchies), который в отличие от K-means более устойчив к зашумленным данным, а благодаря обобщенным представлениям кластеров, скорость кластеризации увеличивается. Алгоритм BIRCH обладает большим масштабированием и возможностью онлайн дообучения.

В ходе первого этапа формируется дерево признаков кластеризации (англ. Clustering Feature). На втором этапе к выявленным кластерам применятся иерархический алгоритм кластеризации.

В результате кластеризации выборка была разбита на 2000 кластеров, а каждый нормированный вектор пакета $\mathbf{z}$ был представлен его кластерной оценкой $\tilde{\mathbf{z}}$ – номером кластера $v = \overline{1,2000}$.

На рисунке 3 представлена визуализация сетевой атаки типа DoS (а) и нормального трафика (б) в виде временного ряда, где по оси Ox – последовательно идущие сетевые пакеты, а по оси Oy – номера кластеров.

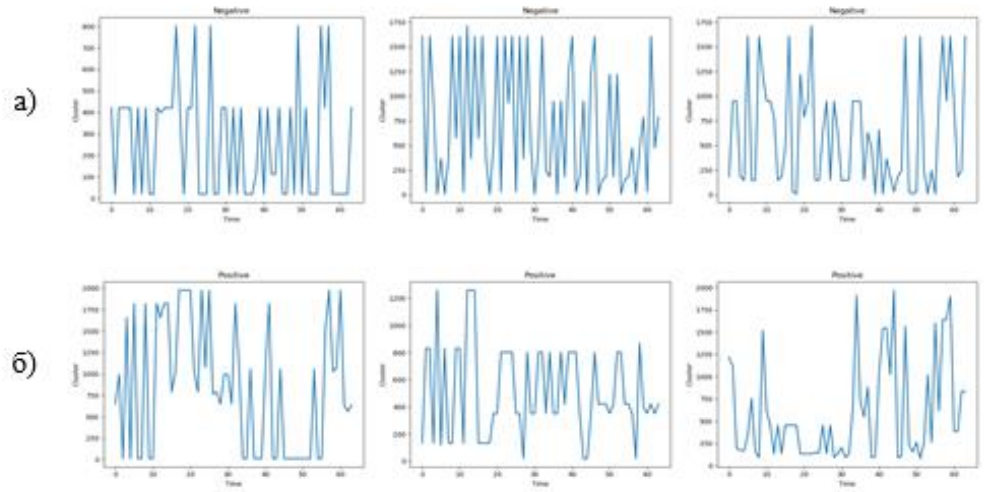

Рис. 3. Визуализация трафика

На рисунке 4 представлен анализ распределения DoS пакетов (а) и пакетов нормального трафика (б) по кластерам, где по оси Ox номер кластера, а по оси Oy количество примеров в кластере.

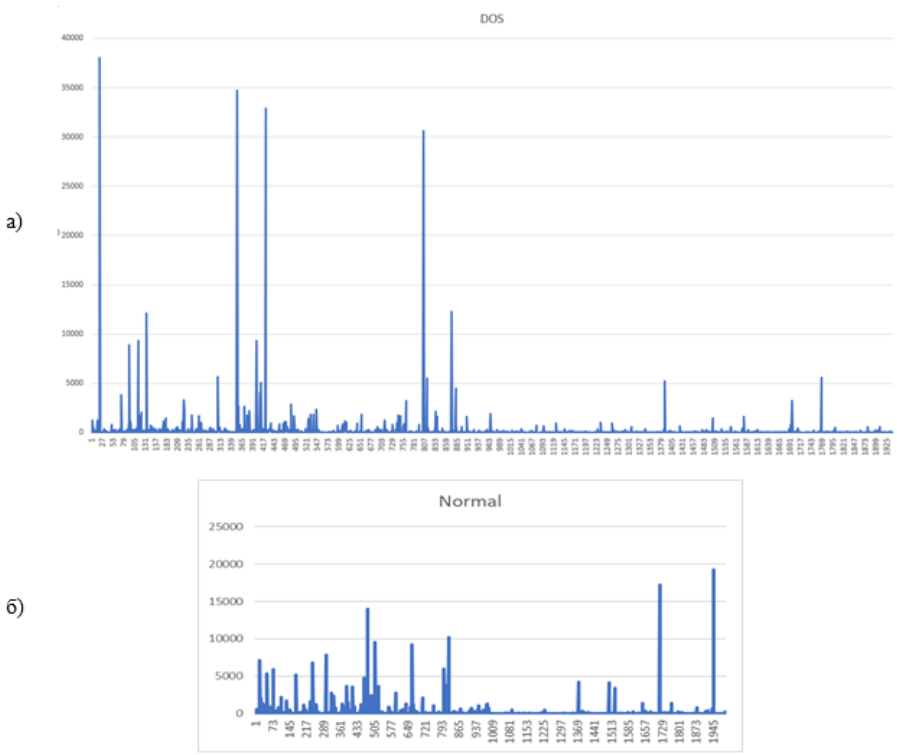

Рис. 4. Распределение пакетов по кластерам: а) DoS атака; б) нормальный трафик

С помощью древовидных моделей машинного обучения можно вычислить важность признака. В частности, с использованием алгоритма градиентного бустинга (англ. Gradient Boosting Machine, GMB) [11], который строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений, нами была проанализирована важность признаков DoS атак в обучающем наборе. В качестве меток классов был установлен номер кластера, что позволило выявить признаки, применяемые алгоритмом кластеризации BIRCH и сравнить их с признаками, выделенными авторами CICIDS2017. На рисунке 5 представлены нормализованные относительные значения важности самых значимых признаков DoS.

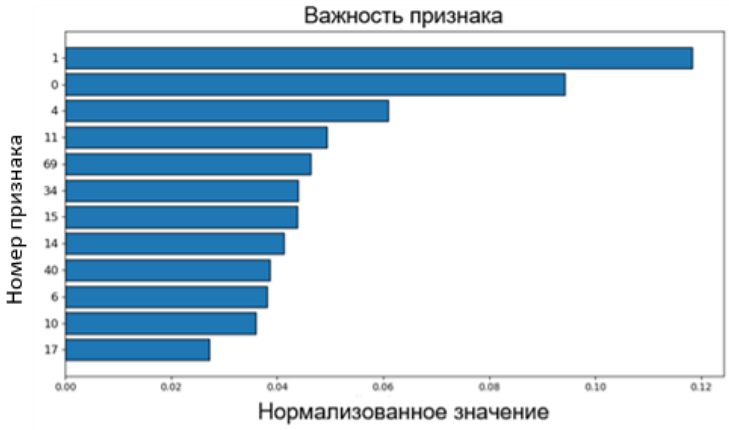

Рис. 5. Важность признаков при кластеризации

Наиболее значимые признаки DoS атак, выделенные при кластеризации: протяженность потока по времени (Flow Duration) (1), порт назначения (Destination Port) (0), общее количество байт потока, переданных в прямом направлении (Total Length of Fwd Packets) (4), минимальное значение длины пакетов в байтах, переданных в обратном направлении (Bwd Packet Length Min) (11), минимальный размер сегмента, переданного в прямом направлении min_seg_size_forward (69).

2.2 Уточнение области интереса

Особенностью сетевой атаки является протяженность во времени, а значит необходимо выполнить анализ последовательности сетевых пакетов и обобщить полученную информацию за определенный временной интервал. Тогда сканирующим окном является последовательность исходных пакетов:

$$(2) \quad \mathbf{W}_t = concat(\mathbf{P}_t, \ldots \mathbf{P}_{t+n}) = concat(\mathbf{z}_t, \ldots \mathbf{z}_{t+n}), \cdots \widetilde{\mathbf{Z}} = [\widetilde{\mathbf{z}}_t, \ldots \widetilde{\mathbf{z}}_{t+n}],$$

где n – количество пакетов, concat – операция конкатенации нескольких подряд идущих кадров в многомерный массив, $\widetilde{\mathbf{z}}_t$ – номер кластера нормированного пакета $\mathbf{z}_t$. Также можно сканировать трафик по последовательности нормированных пакетов или последовательности номеров кластеров.

Согласно временным меткам из датасета CICIDS2017 за 1 секунду проходит около 60-70 пакетов. Тогда примем размер сканирующего окна $n=64$ пакета и смещение $d=10$ пакетов (рисунок 6).

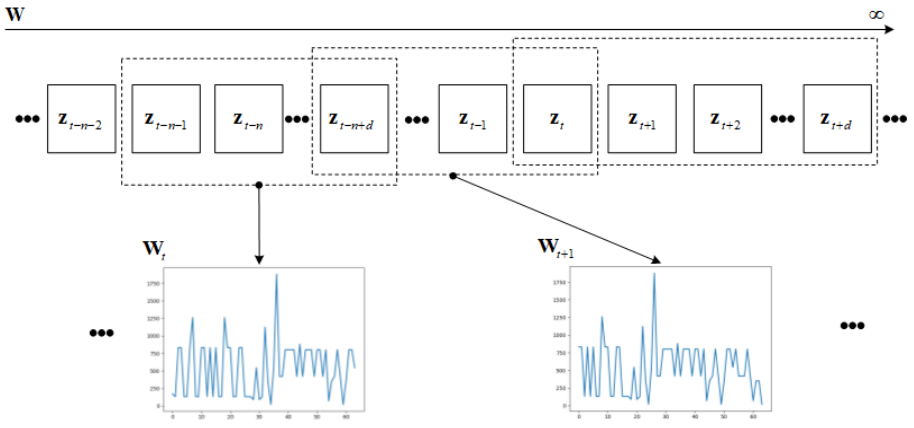

Рис. 6. Сканирующее окно

2.3 Выделение информативных признаков и классификация

Для выделения информативных признаков и вычисления метки класса образа предлагается разработать архитектуру глубокой нейронной сети, реализующей отображение f: $W_t \rightarrow pW_t$, где pW_t – вектор размером $c \times 1$, содержащий вероятности классификации, c – количество классов.

Обучение ГНС производится с использованием обучающего множества, состоящего из одного дня датасета CICIDS2017, который содержит 691 406 пакетов, разбитых на 69 141 окон. Для обучения примем, что пропорции обучающего и валидационного множества составляют 70% на 30% соответственно. Для обучения необходимо присвоить окну $\mathbf{W}_t$ метку класса. Для этого вычисляем долю пакетов, относящихся к DoS атаке (рисунок 7), как сумму всех меток β_k, присутствующих в окне, деленную на размер окна. Если количество DoS пакетов больше 20% будем считать, что окно относится к классу DoS атаки:

$$(3) \quad \beta_k^{\mathbf{W}} = \begin{cases} 1, \text{если } \dfrac{\sum \beta_k}{64} \geq 0{,}2 \\ 0, \text{если } \dfrac{\sum \beta_k}{64} < 0{,}2 \end{cases}.$$

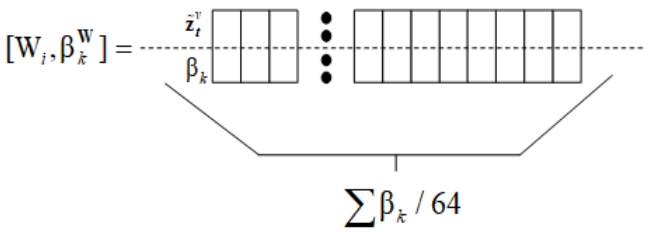

Рис. 7. Вектор для обучения

В качестве классификатора DoS атаки нами разработаны и протестированы 8 различных архитектур ГНС (Рисунок 8), построенных комбинацией слоев свертки, полносвязных слоев (англ. Dense) и рекуррентных слоев LSTM, на вход которых подаются нормированные данные или обработанные различными алгоритмами кластеризации.

Слой свертки осуществляет вычисления следующим образом:

$$\mathbf{h}_{conv}^{l} = \sigma^{ReLU}(\mathbf{h}^{l-1} \cdot \mathbf{K} + b_{conv}^{l}), \quad (4)$$

Где h^{l}_{conv} – выход слоя свертки l, который представляет собой многомерный массив; σ^{ReLU} – функция активации ReLU; b^{l}_{conv} – коэффициент сдвига.

Функция ReLU обнуляет отрицательные элементы вектора $\mathbf{h}$:

$$\sigma^{ReLU}(\mathbf{h}) = \max(0, \mathbf{h}). \quad (5)$$

Выходное значение каждого нейрона $h^{l}_{conn_j}$ полносвязного слоя $\mathbf{h}^{l}_{conn}$ размером $K \times 1$ вычисляется по формуле:

$$h_{conn_j}^{l} = \sigma^{Sigmoid}\left(\sum_i h_i^{l-1} \cdot M_{ij}^{l-1} + b_j^{l-1}\right), \; j = \overline{1,K}, \; i = \overline{1,L}, \quad (6)$$

где h_i^{l-1} – выходной сигнал нейрона предыдущего слоя размером $L \times 1$; b_j^{l-1} – скалярный коэффициент сдвига; $\mathbf{M}$ – матрица весовых коэффициентов размером $i \times j$; $\sigma^{Sigmoid}$ – функция активации Sigmoid.

1) На вход нейронной сети 1 (2LSTM) подается окно из нормализированных векторов пакетов. Архитектура 2LSTM состоит из двух последовательно идущих слоев LSTM и полносвязного слоя.
2) На вход нейронной сети 2 (6CNN1D) подается окно из нормализированных векторов пакетов. Архитектура 6CNN1D состоит из шести последовательно идущих слоев одномерной свертки и полносвязного слоя.
3) На вход нейронной сети 3 (Kmeans-2LSTM) подается окно из кластеризированных оценок векторов пакетов алгоритмом K-means. Архитектура Kmeans-LSTM состоит из слоя встраивания (Embedding), который производит векторизацию входных данных, двух последовательно идущих слоев LSTM и полносвязного слоя.
4) На вход нейронной сети 4 (BIRCH-2LSTM) подается окно из кластеризированных оценок векторов пакетов алгоритмом BIRCH. Архитектура BIRCH-2LSTM состоит из слоя встраивания, двух последовательно идущих слоев LSTM и полносвязного слоя.
5) На вход нейронной сети 5 (Kmeans-6CNN1D) подается окно из кластеризированных оценок векторов пакетов алгоритмом K-means. Архитектура Kmeans-6CNN1D состоит из слоя встраивания, шести последовательно идущих слоев одномерной свертки и полносвязного слоя.
6) На вход нейронной сети 6 (BIRCH-6CNN1D) подается окно из кластеризированных оценок векторов пакетов алгоритмом BIRCH. Архитектура BIRCH-6CNN1D состоит из слоя встраивания, шести последовательно идущих слоев одномерной свертки и полносвязного слоя.
7) На вход нейронной сети 7 (BIRCH-9CNN1D) подается окно из кластеризированных оценок векторов пакетов алгоритмом BIRCH. Архитектура BIRCH-9CNN1D состоит из слоя встраивания, девяти последовательно идущих слоев одномерной свертки и полносвязного слоя.
8) На вход нейронной сети 8 (DUAL-6CNN1D) подается окно из кластеризированных оценок векторов пакетов алгоритмом BIRCH и окно из нормализированных векторов пакетов. Архитектура BIRCH-9CNN1D представлена двумя параллельными ветками, каждая из которых состоит шесть последовательно идущих слоев одномерной свертки и слоя глобальной средней подвыборки. Ветки объединяются слоем конкатенации и полносвязным слоем.

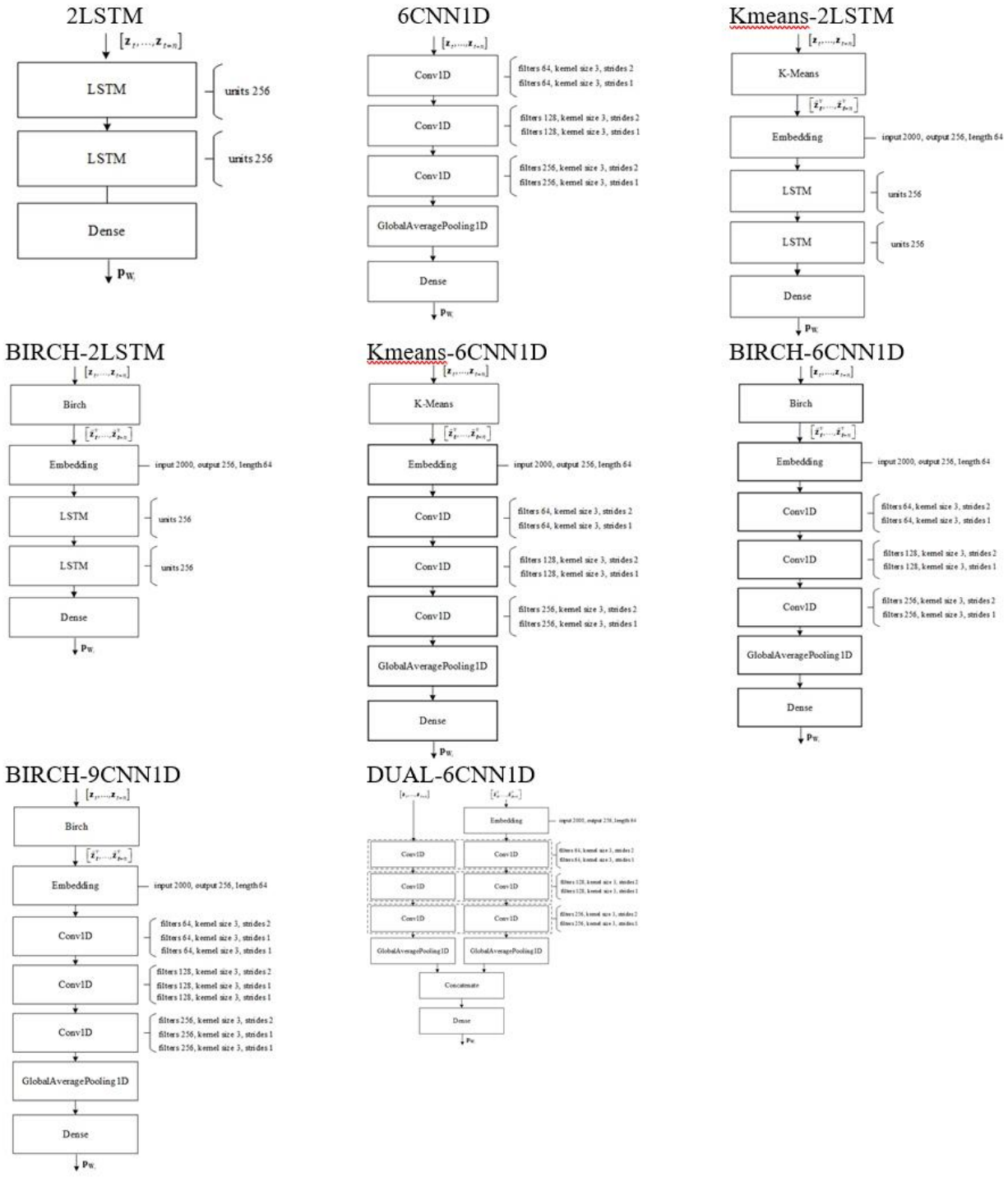

Рис. 8. Архитектуры глубоких нейронных сетей

Задача алгоритма распознавания аномального трафика – выявлять наличие подозрительной активности в сети, тогда выходом всех архитектур является вектор $\mathbf{p}_{\mathbf{W}_t}$, содержащий вероятность DoS атаки. Критерий классификации определяется как $J(\mathbf{f}) = \max \mathbf{p}_{\mathbf{W}_t}$. Если $J(\mathbf{f}) \geq \varepsilon$, где ε – заданный порог, то $\beta_k = \underset{k \in 1..c}{\arg\max}(\mathbf{p}_{\mathbf{W}_t})$, в противном случае классификация считается ошибочной. В задаче обнаружения DoS атак $\varepsilon = 0{,}2$.

2.4 Пример обнаружения аномального сетевого трафика

На рисунке 9 представлена визуализация сетевого трафика с шагом 10 пакетов, где по оси Oy представлен результат классификации окна в виде вероятности обнаружения аномального трафика.

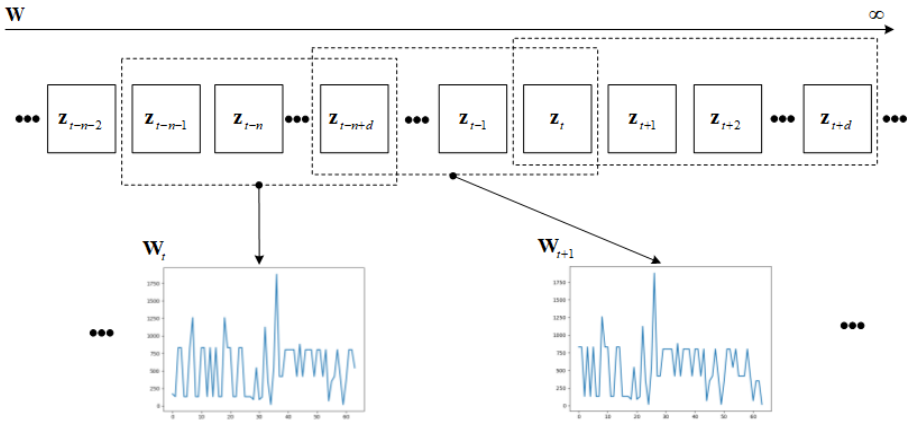

Рис. 9 Результат классификации

С использованием тестирующей выборки, полученной из второго дня датасета CICIDS2017 и содержащей DoS атаки, был проведен эксперимент на оборудовании со следующими параметрами: ЦПУ Intel Core i7-5820K, графический процессор (ГПУ) 1080 Ti. Для обучения каждой архитектуры использовалось 50 эпох.

Размер тестирующей выборки – 225711 последовательных пакетов, разбитых на 22565 сканирующих окон. Были получены следующие результаты (таблица 1) для метрик Precision, Recall F1-score по классам и для Accuracy, AUC – в общем.

Таблица 1. Результат классификации

Показатель / Нейронная сеть	Class	Precision	Recall	F1-score	Accuracy	AUC
2LSTM	0 (9655)	1,00	0,83	0,90	0,92	0,91
	1(12910)	0,89	1,00	0,94		
6CNN1D	0(9655)	0,99	0,99	0,99	0,99	0,99
	1(12910)	0,99	0,99	0,99		
Kmeans-2LSTM	0(9655)	0,95	0,80	0,87	0,90	0,88
	1(12910)	0,87	0,97	0,91		
BIRCH-2LSTM	0(9655)	0,76	0,81	0,79	0,81	0,81
	1(12910)	0,85	0,81	0,73		
Kmeans-6CNN1D	0(9655)	0,99	0,95	0,97	0,97	0,97
	1(12910)	0,96	0,99	0,98		
BIRCH-6CNN1D	0(9655)	0,98	0,96	0,97	0,97	0,97
	1(12910)	0,97	0,99	0,98		
BIRCH-9CNN1D	0(9655)	0,99	0,98	0,98	0,99	0,98
	1(12910)	0,98	0,99	0,99		
DUAL-6CNN1D	0(9655)	0,99	0,99	0,99	0,99	0,99
	1(12910)	0,99	1,00	0,99		

Рекуррентные сети LSTM зарекомендовали себя при обработке текстовых последовательностей естественного языка (системы аннотирования, перевода и др.), однако с задачей анализа временного ряда сетевого трафика LSTM сеть справляется довольно плохо – точность 91%.

Глубокая сверточная нейронная сеть позволяет находить скрытые зависимости в данных, однако такие архитектуры трудно интерпретируемы. Применение кластеризации позволяет выявить важные признаки и снизить размерность входного вектора. Использование кластеризации на ГПУ позволит получить существенный прирост скорости обработки.

Стоит отметить, что для рекуррентных сетей предпочтительней использовать алгоритм K-means. Точность классификации 88% в отличие от 81% при использовании BIRCH.

Для сверточных сетей BIRCH дает более высокий результат классификации – 97%. Увеличение количества сверточных слоев повышает точность классификации до 98%, но усложняет модель в части вычислений. Дуальная архитектура сети позволяет повысить точность классификации до 99% путем подкрепления входного вектора меткой его кластера.

Применение кластеризации в качестве предобработки для глубоких архитектур нейросетей, работающих на ГПУ, позволяет значительно повысить скорость обработки и делает возможным обнаружение DoS атак в высоконагруженных корпоративных сетях. Скорость обработки 32 768 окон, соответствующих 512 с трафика, составляет от 1,8 с до 2 с, что достаточно для работы в реальном времени.

Заключение

Дана постановка задачи распознавания аномалий сетевого трафика.

Проанализированы признаки датасета CICIDS2017 и проведена их нормализация.

Для распознавания аномалий сетевого трафика предлагается вычислительный метод с использованием глубоких нейронных сетей и методов кластеризации. Экспериментально показано, что применение кластеризации совместно со сверточными слоями позволяет достигнуть хороших результатов в режиме реального времени при решении задач информационной безопасности. За счет применения операции свертки существенно снижается количество настраиваемых параметров по сравнению с традиционными НС, а чередование сверточных слоев позволяет выстроить иерархию признаков и с достаточно высокой скоростью и точностью распознавать начало атаки. С использованием дуальной сети была получена точность классификации 99,26%.

Применение архитектур глубоких нейронных сетей позволяет производить вычисления на графическом процессоре и на нейронных модулях типа Neural Compute Stick.

Реализован подход для обнаружения сетевой атаки типа DoS. В отличие от предлагаемых ранее подходов в работе применяется сканирующее окно, размерность которого снижена путем кластеризации данных.

В дальнейшем предполагается перенести предложенный подход на аппаратную вычислительную базу для реализации интеллектуального межсетевого экрана.

Перспективным направлением исследования является реализация многоклассовой классификации, алгоритма реакции и предотвращения последствий атаки с использованием алгоритмов нечеткой логики.

Благодарности

Работа выполнена при поддержке Минобрнауки России научного проекта – госзадания в рамках проектной части № 2.1898.2017/ПЧ "Создание математического и алгоритмического обеспечения интеллектуальной информационно-телекоммуникационной системы безопасности вуза".

Литература

1. *Riyazahmed A.J.* Network Intrusion Detection System Using Machine Learning // Indian Journal of Science and Technology Vol. 11. 2018, №48. – P. 1-6 .
2. *Амосов О.С., Магола Д.С., Баена С.Г.* Сетевая классификация атак в задачах информационной безопасности на основе интеллектуальных технологий, фрактального и вейвлет-анализа // Ученые записки КнАГТУ. 2017. № IV-1 (32). – С. 19-29.
3. *Амосов О.С., Муллер Н.В.* Исследование временных рядов с применением методов фрактального и вейвлет анализа // Интернет-журнал Науковедение. 2014. №3(22). – С. 89.
4. *Resende P.A.A., Drummond A.C.* Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling // Security Privacy Vol. 1. 2018, №4. – P. 1-13.
5. *Haripriya A.P., Kulothungan K.* Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things // EURASIP Journal on Wireless Communications and Networking Vol. 2019. 2019, №1. -P 90-105.
6. *Sharafaldin I., Habibi Lashkari A., Ghorbani A.* Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization // In Proceedings of the 4th International Conference on Information Systems Security and Privacy Vol. 1. -P 108-116.
7. *Chockwanich N., Visoottiviseth V.* Intrusion Detection by Deep Learning with TensorFlow // 21st International Conference on Advanced Communication Technology (ICACT). 2019. – P. 654-659.

8. *Амосов О.С., Магола Д.С., Пащенко Ф.Ф., Амосова С.Г.* Классификация сетевых атак на основе глубоких нейронных сетей с 1D-сверточными и рекуррентными слоями // XIII Всероссийское совещание по проблемам управления, ВСПУ-2019, Москва, 17-20 июня 2019 г. – С. 1-5.
9. *Амосов О.С.* Фильтрация Марковских последовательностей на основе байесовского, нейросетевого подходов и систем нечеткой логики при обработке навигационной информации // Известия Российской академии наук. Теория и системы управления. 2004, № 4. – С. 61-69.
10. *Zhang T., Ramakrishnan R., Livny M.* BIRCH: An Efficient Data Clustering Method for Very Large Databases // Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. 2019. -P 103-114
11. *Friedman J.H.* Stochastic gradient boosting // Computational Statistics and Data Analysis Computational Statistics & Data Analysis Vol. 38, Issue 4, 2002, P. 367-378