УДК 004.032.26

Вестник УрФО № 3(33) / 2019, с. 27–42

Ермаков Р. Н.

DOI: 10.14529/secur190304

ДЕТЕКТИРОВАНИЕ СЕТЕВЫХ ПРОТОКОЛОВ С ПРИМЕНЕНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ И АЛГОРИТМОВ НЕЧЕТКОЙ ЛОГИКИ В СИСТЕМАХ АНАЛИЗА ТРАФИКА

В статье рассматривается новый эффективный подход к анализу сетевого трафика с целью определения протокола информационного обмена прикладного уровня. Дается краткое описание структуры алгоритма классификации сетевых пакетов на принадлежность к одному из известных сетевых протоколов. Для определения протокола используется принцип высокоскоростной однопакетной классификации, который заключается в том, что анализируется информация, передаваемая в каждом конкретном пакете. Используются элементы поведенческого анализа, а именно, классифицируются переходные состояния протоколов информационного обмена, что позволяет достичь более высокого уровня верности классификации и более высокой степени обобщения на новых тестовых выборках. Применяются алгоритмы нечеткой логики и нейронные сети. Демонстрируются результаты тестирования построенного программного модуля, способного идентифицировать сетевые протоколы информационного обмена.

Ключевые слова: *классификация сетевых пакетов, искусственные нейронные сети, логистическая регрессия, анализ сетевого трафика, глубокий анализ пакетов.*

Ermakov R. N.

DETECTION OF NETWORK PROTOCOLS WITH APPLICATION OF MACHINE LEARNING METHODS AND FUZZY LOGIC ALGORITHMS IN TRAFFIC ANALYSIS SYSTEMS

This article presents a new effective approach to analyzing network traffic in order to determine the protocol of information exchange. A brief description of the structure of the algorithm for classifying network packets by belonging to one of the known network protocols is given. To define the protocol, the principle of high-speed one-packet classification is used, which consists in analyzing the information transmitted in each particular packet. Elements of behavioral analysis are used, namely, the transition states of information exchange protocols are classified, which allows to achieve a higher level of accuracy of classification and a higher degree of generalization in new test samples. Fuzzy logic algorithms and neural networks are used. The test results of the constructed software module capable of identifying network protocols for information exchange are demonstrated.

Keywords: *network packet classification, artificial neural networks, logistic regression, machine learning, network traffic analysis, in-depth packet analysis.*

1. Введение

В последние годы устойчиво возрастает интерес операторов телекоммуникационного рынка к системам анализа сетевого трафика (NetworkTrafficAnalysis– NTA, DeepPacketInspection – DPI) и к системам обеспечения комплексной информационной безопасности. При это предполагается, что и в дальнейшем его рост будет продолжаться, особенно для систем обеспечения информационной безопасности, использующих современные интеллектуальные математические модели и методы. Анализ трафика в течение многих лет остается актуальным направлением исследований. Этому способствуют две основные причины: рост трафика (в том числе вредоносного) и появление новых технологий.

Разработано и эксплуатируется множество систем обеспечения информационной безопасности. К таким системам можно отнести:

• системы управление правами доступа (IdentityManagement – IDM);

• системы контроля действий администраторов (PrivelegeAccountsManagement – PAM);

• развитые межсетевые экраны (Next Generation Firewalls – NGFW);

• средства анализа защищенности (Security Information and Event Management – SIEM);

• антивирусные решения (Antivirus, Antibot, Malware Protection – AV);

• системы обнаружения вторжений и аномалий (Intrusion Detection System – IDS, Application protocol-based IDS – APIDS);

• системы предотвращения атак (Intrusion Prevention System – IPS);

• системы аудита и мониторинга средств безопасности (Security Information and Event Management – NMS);

• системы защиты от атак класса «Отказ в обслуживании» (DDoS Protection Systems – DDoS PS);

• системы управления политиками сетевого трафика (Policy and Charging Enforcement Function – PCEF, Policy and Charging Rules Function – PCRF, Network Access Control – NAC);

• другие системы.

Системы анализа трафика NTA являются необходимым инструментом многих представленных классов других систем обеспечения информационной безопасности, таких как IDS, IPS, NMS, DDoS PS и другие. Решение задачи классификации сетевых протоколов инфокоммуникационного обмена позволяет решать следующие типы задач:

• разработка датчиков обнаружения атак [1];

• идентификация типов устройства [2], задействованных в информационном обмене;

• определение типовых приложений, запущенных на устройствах, задействованных в процессе информационного обмена;

• создание датчиков обнаружения аномального или поддельного трафика (в случае выдачи одного протокола за другой протокол);

• классификация сетевых приложений прикладного уровня функционирующих на седьмом уровне модели OSI (SKYPE, Facebook, Viber, Telegram и др.).

В настоящее время в целях обеспечения информационной безопасности ведутся обширные исследования и поиск новых способов определения DDoS-атак. Как правило, эти способы опираются на выявление сетевых активностей и аномалий [3]. Подобные задачи можно также эффективно решать и с использованием классификатора сетевых пакетов (КСП) прикладного уровня. Однако стоит принять во внимание, что это несколько медлительный способ. Вначале с использованием КСП могут быть классифицированы протокол информационного обмена и типы устройств, задействованные при обмене. Далее могут быть выявлены приложения на устройствах информационного обмена и затем сетевые приложения прикладного уровня. Такой способ определения потенциальных сетевых угроз не является высокоскоростным. Однако он может быть очень эффективным в рамках тестовой среды для проведения полноценных комплексных исследований в задачах идентификации DDoS-атак.

Вместе с тем, классификатор сетевых пакетов (КСП) прикладного уровня может быть очень полезен при распознавании внутреннего состояния, в котором может находиться тот или иной протокол в процессе информационного обмена на этапе handshake (рукопожатия), что является важным элементом поведенческого анализа.

Классификация трафика может быть осуществлена на основе:

• использования традиционных подходов к анализу трафика (сигнатурный и поведенческий) в зависимости от того, зашифрован трафик или нет [4]. Сигнатурный метод основан на анализе номеров портов пакетов, сигнатуры протокола (payload-based). Поведенческий анализ способов обмена пакетами между абонентами и свойств сетевого трафика основан на исследовании последовательности размеров пакетов, временных интервалов между пакетами и т. д.;

• применения известных математических подходов:

a) базовые статические алгоритмы;

b) машинное обучение (метод опорных векторов и метод случайных лесов [5]);

c) алгоритмы нечеткой логики;

d) методы теории нейронных сетей;

• применения новых математических моделей и алгоритмов (исследуется в настоящей работе).

Качественная классификация сетевых протоколов прикладного уровня, как в плане ее верности и степени обобщения на новых тестовых выборках, так и в плане уменьшения требований к вычислительной мощности, оказывает важное влияние на функционирование систем NTA [6, 7], анализа трафика DPI [8], IDS/IPS [9], DDoS PS [10] и других, как на весь технологический процесс, так и на качество анализа.

По уровню классификации сетевых пакетов различают: «поверхностный» анализ пакетов (SPI – ShallowPacketInspection), «средний» анализ пакетов (MPI – MediumPacketInspection) и «глубокий» анализ пакетов (DPI – DeepPacketInspection). Анализаторы «поверхностного» уровня функционируют в простейших межсетевых экранах, где решение о блокировании того или иного пакета обычно принимается в соответствии со списком запрещенных IP-адресов и номеров портов. Программные средства анализа трафика, относящиеся к «среднему» уровню, позволяют проводить фильтрацию трафика с использованием информации о формате передаваемых данных, а также более полной локализации отправителя. Эти инструменты обычно выступают в роли посредника (proxy) между про-

вайдером доступа к Интернет и внутренней сетью. Системы «глубокого» анализа пакетов предназначены для идентификации приложений, участвующих в сетевых взаимодействиях. Поэтому «углубленный» разбор предполагает анализ содержимого сетевых пакетов всех уровней и назначение подобных систем — это обеспечение информационной безопасности и мониторинг качества каналов связи.

Как правило, для анализа сетевого трафика исследователи в своих работах определяют протокол прикладного уровня с использованием алгоритмов машинного обучения «с учителем» [5, 11, 12]. В [5] задача классификации решалась методом опорных векторов и с помощью алгоритма «случайных лесов». Результаты исследований на тестовых выборках в этой работе показали, что оба подхода приводят к высокой верности классификации, 98% и выше. Однако ничего не сказано о среднем времени классификации пакетов алгоритмами на конкретных аппаратных платформах и операционных системах. Остается вопрос: можно ли использовать такие «тяжеловесные» алгоритмы классификации в реальных системах анализа трафика с учетом предъявляемых требований по вычислительной производительности? Для классификации сетевых пакетов в настоящей работе использовались алгоритмы нечеткой логики (модель Мамдани) и методы машинного обучения, в частности, нейронные сети, а именно, логистическая регрессия.

Разработка КСП состояла из четырех этапов: (1) мониторинг и сбор пакетной статистической информации наиболее известных протоколов сетевого трафика, (2) предобработка первичной пакетной статистической информации, (3) построение классификатора сетевых пакетов и (4) тестирование.

2 Постановка задачи классификации трафика

Постановка задачи классификации трафика формулируется следующим образом. Имеется множество исследуемых объектов IP-пакетов прикладного уровня:

$$P = \{P_1, P_2, ..., P_w\} \quad (1)$$

где P_w – анализируемый пакет из последовательности пакетов (трафика).

Каждый объект (IP-пакет) характеризуется набором переменных (атрибутов):

$$P_w = \left\{X_1^w, X_2^w, \cdots, X_{10}^w,\ H_j^w, H_{j+n}^w, H_{j+n+\cdots+k}^w\right\}, \quad (2)$$

$$Z = \{Z_1, Z_2, Z_3\}, \quad (3)$$

где, X_n^w–наблюдаемый *n*-атрибут *w*-пакета, область допустимых значений которого содержится в RFC (Requestforcomments), H_j^w– байтовая последовательность *Payload_hex* размером – J, Z– зависимое множество, которое необходимо определить. Множество Z включает: тип протокола – Z_1, вероятность принадлежности к выявленному типу протокола – Z_2, внутренне состояние протокола в процессе информационного обмена – Z_3.

При этом каждая переменная X_n принимает значение из некоторого множества:

$$X_n = \{Xn_1, Xn_2, ..., Xn_M\}, \quad (4)$$

где, Xn_M варианты значений атрибутовиз Мвозможных вариантов, описанных в RFC.

Таким образом, задача классификации сетевого протокола сводится к определению множества *Z* на основе значений атрибутов последовательности пакетов.

Решая прикладную задачу классификации, с учетом анализа исследований [8, 13] в настоящей работе был выделен следующий набор атрибутов:

X_1 – *EtherType* (тип стандарта протокола Ethernet);

X_2 – *SourceIPAddress* (IP-адресу отправителя);

X_3 – *DestinationIPAddress* (IP-адрес получателя);

X_4 – *Multicast* (принимает значение 1, если multicast, в противном случае 0);

X_5 – *IPProtocol* (тип транспортного уровня);

X_6 – *PacketLength* (длина сетевого пакета в байтах);

X_7 – *SourcePort* (порт TCP/UDP отправителя);

X_8 – *DestinationPort* (порт TCP/UDP получателя);

X_9 – *Hex_length* (количество байт в шестнадцатеричной последовательности– HexStream, которая является частью полезной нагрузки – payload);

X_{10} – *Payload_type* (атрибут для обеспечения обучения моделей классификации сетевых пакетов по схеме с учителем);

X_{11} – *Payload_hex* (передаваемая шестнадцатеричная последовательность в контенте пакета протокола прикладного уровня часть payload размером J), для удобства в тексте статьи в ряде случаев обозначается как H.

Также был включен дополнительный параметр - маркировка класса Payloadtype, чтобы иметь возможность обучения моделей классификации сетевых пакетов по схеме с учителем.

Вариант исходных данных трафика может иметь вид, как показано в табл.1.

Таблиwа 1

Представление варианта последовательности пакетов в сетевом трафике

N	Ethernet Type	Source IP Address	Destination IP Address	Mult-cast	IP proto	Packet length	Src port	Dst port	type proto	Hex length	Hex stream
0	0x800	172.16.0.1	172.16.0.10	0	6	73	53986	21	FTP	7	504153
1	0x800	172.16.0.1	172.16.0.10	0	6	72	53986	21	FTP	6	4c4953
2	0x800	172.16.0.1	172.16.0.10	0	6	90	53986	21	FTP	24	504f52
3	0x800	192.168.10.3	192.168.10.8	0	6	91	443	61983	TLSv1	37	150301
4	0x800	13.79.241.1	192.168.10.1	0	6	459	443	61867	TLSv1	405	170303
5	0x800	192.168.10.3	192.168.10.8	0	6	91	443	61986	TLSv1	37	150301
6	0x800	94.100.181.5	192.168.10.5	0	6	491	443	61662	TLSv1	437	170301
7	0x800	192.168.10.3	192.168.10.8	0	6	199	443	61993	TLSv1	145	170303
8	0x800	192.168.10.3	192.168.10.8	0	6	113	62544	443	TLSv1	59	170303
9	0x800	192.168.10.1	13.79.241.1	0	6	267	61988	443	TLSv1.2	213	d2a6ea
10	0x800	13.79.241.1	192.168.10.1	0	6	571	443	61990	TLSv1.2	517	6edb93
11	0x800	13.79.241.1	192.168.10.1	0	6	731	443	61988	TLSv1.2	677	170303
12	0x800	192.168.10.7	40.115.1.4	0	6	305	61991	443	TLSv1.2	251	170303
13	0x800	40.115.1.4	192.168.10.7	0	6	1223	443	61991	TLSv1.2	1169	05bb0f
14	0x800	192.168.10.7	40.115.1.4	0	6	268	61991	443	TLSv1.2	214	170303

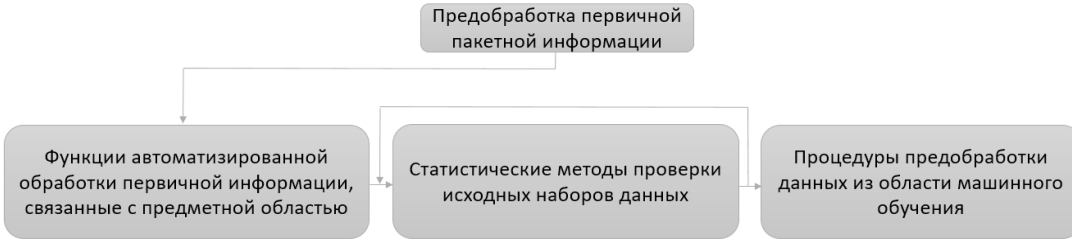

Рис.1 Интеллектуальная автоматизированная предобработка первичных пакетных данных

3. Мониторинг и сбор пакетной статистической информации.

Мониторинг и сбор пакетной статистической информации наиболее часто встречающихся протоколов сетевого трафика (TLS v1, TLS v1.2, SSH v2, HTTP, FTP и др.) был осуществлен с использованием открытого программного обеспечения Wireshark [14] и включал в себя решение следующих задач:

а) выбор наиболее подходящих входных переменных для построения модели классификации сетевых пакетов;

б) формирование набора первичных репрезентативных выборок -DUMP в PCAP-формате с пакетной информацией по указанным выше протоколам (объемом трафика ~1Гб);

в) автоматическое формирование вторичных выборок для анализа.

4. Предобработка первичной пакетной статистической информации

Предобработка первичной пакетной информации состоит из трех последовательно выполняющихся процедур, которые представлены на рис. 1.

В первом блоке предобработки задействован следующий функционал:

• проверка payload на предмет шифрованной контентной информации (потребовало реализации отдельного специального технологического ПО на языке Python);

• обработка payload для лучшего визуального восприятия и возможности эвристического (поведенческого) анализа;

• разделение полученных классифицируемых сетевых пакетов на однородные группы (A, B, C, D) на основе значений некоторых входных признаков (тип стандарта протокола Ethernet, Multicast и тип транспортного уровня), а также формирование тестовых и тренировочных наборов.

В таблице 1 приведен пример тестового плоского файла в csv-формате с пакетной статистической информацией.

Разделение классифицируемых сетевых пакетов на принадлежность к протоколам (DHCPv6, DNS, FTP, HTTP и др.) на группы (, , ,) производится на основе следующих логических правил:

$$
\begin{aligned}
GROUP_A &= if\ (\text{ethertype} == \text{IPv4})\&\&(Multicast == 0)\ \&\&(IP\_PROTO == \text{TCP});\\
GROUP_B &= if\ (\text{ethertype} == \text{IPv4})\&\&(Multicast == 0)\ \&\&(IP\_PROTO == \text{UDP});\\
GROUP_C &= if\ (\text{ethertype} == \text{IPv4})\&\&(Multicast == 1)\ \&\&(IP\_PROTO == \text{UDP});\\
GROUP_D &= if\ (\text{ethertype} == \text{IPv6})\&\&(Multicast == 1)\ \&\&(IP\_PROTO == \text{UDP});
\end{aligned}
\tag{5}
$$

В результате вычислений по выражению (4), классифицируемые сетевые пакеты распределяются по группам для идентификации соответствующих протоколов информационного обмена:

$$
\begin{aligned}
GROUP_A &= \{\text{TLSv1}, \text{TLSv1.2}, \text{TCP}, \text{SSHv2}, \text{HTTP}\};\\
GROUP_B &= \{\text{UDP}, \text{STUN}, \text{QUIC}, \text{NBNS}, \text{DNS}, \text{BROWSER}\};\\
GROUP_C &= \{\text{SSDP}, \text{MDNS}, \text{LLMNR}\};\\
GROUP_D &= \{\text{SSDP}, \text{MDNS}, \text{LLMNR}, \text{DHCPv6}\};
\end{aligned}
\tag{6}
$$

Попавшие в одну группу протоколы будем считать во многом схожими, а общую выборку в группе – однородной.

Внутри каждой группы данные разделяются на тренировочные и тестовые наборы данных.

На современном этапе развития математического моделирования принято считать, что репрезентативные исходные наборы данных во многом обеспечивают конечный успех всего моделирования – получение адекватных моделей. Как правило, проводятся статистические проверки исходных данных, выявляются и исключаются из тренировочных выборок сильно зашумленные или избыточные исходные данные. Во втором блоке алгоритма предобработки выделяются хорошие наборы данных для дальнейшего построения моделей классификации сетевых пакетов прикладного уровня.

В третьем блоке алгоритма используются следующие процедуры предобработки данных, как правило, применяемых в методах машинного обучения:

1) *обработка категориальных данных,*

2) *масштабирование признаков*, которое включает в себя приведение разных признаков к одинаковой шкале (на практике существуют два общих подхода к приведению разных признаков к одинаковой шкале:

- нормализация;
- стандартизация,

3) *отбор содержательных признаков*.

После завершения процедуры нормализации входных переменных, некоторые из которых далее проходят процедуру фаззификации и преобразуются в нечеткие индикаторы, характеризуемые лингвистическими переменными (ЛП). Нечеткий индикатор – это число в диапазоне [0, 1], которое характеризует оценку показателя использующегося в качестве атрибута.

В основу нечеткого индикатора положена оценка эксперта, которая моделируется функцией принадлежности, при этом носителем выступает допустимое множество значений анализируемого показателя.

В общем случае могут также применяться различные модели функции принадлежности, например, такие как треугольная, либо трапециевидная или обобщенная колоколообразная (Рис. 2).В данном исследовании применялись треугольная, трапециевидная, П-образная функции принадлежности и синглетон.

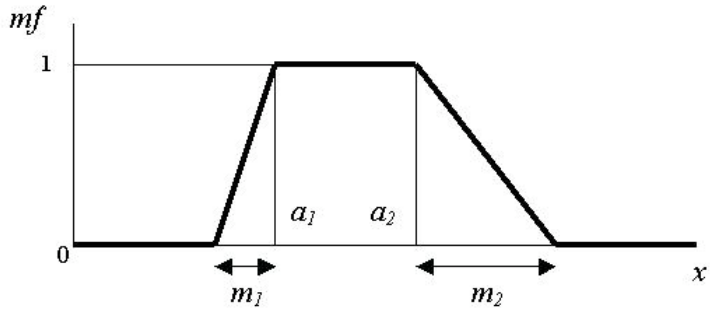

Рис 2.Трапециевидная функция принадлежностиmf нечеткому множеству

5. Построение классификатора сетевых пакетов

В настоящее время идея совместного использования нейросетевого подхода классификации сетевых пакетов и анализ DPI, NTA обсуждаемая исследователями и специалистами [15].

Следует отметить, что ряд входных признаков из числа выделенных и представленных выше являются категориальными. Другие являются количественными. Учитывая, значительное количество анализируемых данных для оптимизации вычислений требуется использовать декомпозицию (свертку).

Для этой цели специалисты машинного обучения часто применяют линейный дискриминантный анализ (LDA) [16] и/или глубокие сверточные нейронные сети[17]. В нашей работе для этой цели используются модели на основе логических правил и алгоритмов нечеткой логики (применялся алгоритм нечеткого вывода Мамдани [18-20]). Применение предложенного нами способа позволяет оперировать нашими знаниями об объекте исследования, например, с использованием RFC-описаний сетевых протоколов.

При реализации «глубокого» анализа пакетов в настоящей работе рассматривается комбинированный метод классификации трафика, основанный на применении теорий нейронных сетей и нечетких множеств. При этом существенный выигрыш в классификации трафика получен при двухэтапном решении задачи, включающем:

– первый этап включает в себя - выполнение процедуры снижения размерности входного пространства признаков (свертки);

– второй этап завершает классификацию трафика с применением логистической регрессии «с учителем» или с использованием алгоритма нечеткого вывода Мамдами.

5.1 Метод однопакетной классификации сетевых пакетов на основе алгоритмов нечеткой логики и нейронных сетей

5.1.1 Особенности алгоритма нечеткого вывода Мамдани

Алгоритм нечеткого вывода Мамдани задействовался в общей схеме классификации сетевых пакетов в настоящей работе. Рассмотрим основные особенности данного алгоритма.

Целевой функцией будем считать выполнение такого отображения своих входов (вектор X) в выход Y, которое обеспечивало бы как можно более точную аппроксимацию реальной системы, например, в смысле средней абсолютной погрешности.

Указанное отображение предполагает существование некоторой геометрической поверхности, которую принято называть поверхностью отображения, в пространстве, задаваемым декартовым произведением [21].

Алгоритм Мамдани представляет собой множество правил, где каждое правило задает в указанном пространстве некоторую нечеткую точку. На основе множества нечетких точек формируется нечеткий график, механизм интерполяции между точками, который зависит от используемого аппарата нечеткой логики.

Формально алгоритм Мамдани может быть определен следующим образом:

• формирование базы правил систем нечеткого вывода;

• фаззификация входных переменных;

• агрегирование подусловий в нечетких правилах продукций, при этом для нахождения степени истинности каждого из правил нечетких продукций используются парные нечеткие логические операции, те правила, степень истинности которых отлична от нуля, считаются активными и используются для дальнейших расчетов;

• активизация подзаключений в нечетких правилах продукций, которую также часто называют выводом на правилах, выполняется с использованием операторов нечеткой импликации, например, если вывод следует осуществлять в соответствии с правилом: ЕСЛИ *(x = A1)* ТО *(y = B1),* используя импликацию Мамдани, можно определить активизированную функцию принадлежности импликации $A \rightarrow B$, которая представляет собой некоторое нечеткое отношение

$$R\colon \mu_R(x,y) = MIN\big(\mu_A(x), \mu_B(y)\big), \quad (7)$$

$R\colon A \rightarrow B$надекартовопроизведение $X \times Y$;

• аккумуляция заключений (процесс определения общего вывода) нечетких правил продукций наиболее часто выполняется так, как показано на следующем примере, пусть дана нечеткая модель с базой правил вида:

$$\begin{aligned} &R1\text{: ЕСЛИ } (x = A1) \text{ ТО } (y = B1),\\ &R2\text{: ЕСЛИ } (x = A2) \text{ ТО } (y = B2), \end{aligned} \quad (8)$$

при этом требуется определить результирующую функцию принадлежности $\mu_{res}(y)$ вывода из всей базы правил, тогда все правила, входящие в базу, можно объединить в одно составное правило следующего вида:

$$R\text{: ЕСЛИ } (x = A1) \text{ ТО } (y = B1) \text{ ИЛИ ЕСЛИ } (x = A2) \text{ ТО } (y = B2), \quad (9)$$

это означает, что правило R состоит из двух простых правил R1 и R2, объединенных логической связкой ИЛИ, которое может быть представлено и так:

$$R = R1 \cup R2 \quad (10)$$

поскольку каждое правило представляет собой нечеткое отношение двух аргументов (импликацию), результирующее отношение R можно найти с использованием одной из *s*-норм, например, оператора *MAX*, его функцию принадлежности $\mu_R(x,y)$ можно получить на основе функций принадлежности состав-

ляющих его отношений (импликаций) по формуле (11):

$$\mu_R(x,y) = MAX(\mu_{R1}(x,y), \mu_{R2}(x,y)), \quad (11)$$

• дефаззификация входных переменных, которая, как правило, задействует популярный *метод центра тяжести*, однако мы применили также очень эффективный метод дефаззификации, называемый *методом высот* (Рис.3), расчеты по которому производятся по формуле (12):

$$y^* = \frac{\sum_{j=1}^{m} y_j \mu_{B_j^*}(y)}{\sum_{j=1}^{m} \mu_{B_j^*}(y)}, \quad (12)$$

где m – число правил.

Представленный на рис.6 алгоритм производит расчеты с использованием искусственных нейронных сетей (см. раздел 5.1.3).

В настоящее время архитектура нейронной сети, как правило, определяется на основе одного из общепринятых подходов:

1) с использованием языка программирования Python с его богатым и быстро обновляющимся арсеналом библиотек, связанных с методами машинного обучения (Scikit-learn [26], Keras [27], Spark [28,29], TensorFlow[30-32], Theano[17]);

2) с использование языка программирования C/C++ и специализированных открытых библиотек, в которых реализованы многие алгоритмы машинного обучения, например, OpenCV [33,34].

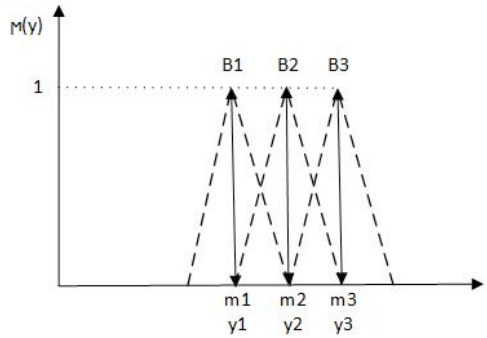

Рис. 3. Замена нечетких множеств Bj одноэлементными множествами (синглетонами)

Таким образом, основными достоинствами алгоритма являются простой способ отображения входных данных (вектор X) в выход Y для достижения высокой точности апроксимации реальной системы при минимальной вычислительной нагрузке и возможность построения эффективных нейронечетких сетей, что предопределило наш интерес к алгоритму нечеткого вывода Мамдани в части его применимости в задачах определения протоколов информационного обмена прикладного уровня на основе классификации сетевых пакетов.

5.1.2 Искусственные нейронные сети

С середины 2000-х годов отмечается возрождение интереса к теме, связанной с практическим применением искусственных нейронных сетей в задачах распознавания образов, аппроксимации функций, обработка сигналов и др. Этому способствовали разработанный прорывной алгоритм быстрого обучения, предложенный Дж. Хинтоном [22], и более позднее появление графических процессоров для параллельных численных расчетов (примерно в 2011 году).

Нейросетевой классификатор – это, по сути, геометрический классификатор, с построения которого начинается вся современная литература по машинному обучению, например, в работах [23-25].

Эти подходы предполагают построение интеллектуальных решений с использованием доступных современных аппаратных платформ, таких какx64-86 (CPU). Однако для других появляющихся и совершенствующихся аппаратных платформ и технологий, таких как Эльбрус [35], графические ускорители вычислений GPU и др., находится место собственным реализациям нейросетевых решений с применением любого языка программирования высокого уровня (C/C++, Python и др.).

В данной работе использовалась собственная реализация нейросетевого решения (логистическая регрессия), разработанная на C++. Вместе с тем, очень важно отметить, что геометрические классификаторы во многих случаях оказываются весьма полезными и эффективными, особенно в задачах определения протоколов информационного обмена и приложений прикладного уровня.

Возвращаясь к выбранной в данной работе архитектуре нейронной сети (многослойная сеть прямого распространения с одним скрытым слоем, который включает 12

нейронов для модели классификации сетевых пакетов на соответствие протоколу TLSv1 и, соответственно, 11 нейронов на соответствие протоколам TLSv1.2 и HTTP), напомним, что при решении большинства прикладных задач полученная взвешенная сумма входных сигналов (см. рис. 4) преобразуется в выход нейрона с помощью некоторой нелинейной функции σ, не обладающей памятью. Данную функцию принято называть активационной:

$$y = \sigma(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_n x_n)$$
$$\sigma(x) = \frac{1}{1+e^{-x}}; \sigma(x) = \frac{e^{-x}}{(1+e^{-x})^2} \quad (14)$$

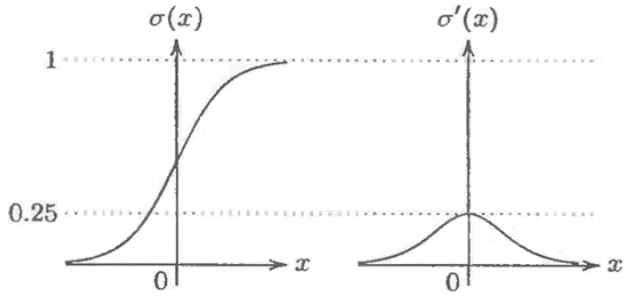

Рис. 4. Сигмоида и ее производная

Выбор активационной функции зависит от специфики решаемой прикладной задачи, часто используют сигмоидную функцию (сигмоиду). В данной работе в качестве активационной функции также применялась сигмоида. Сигмоида и ее производная, показанные на рис. 4, имеют следующий вид:

Методы градиентного спуска и масштабируемых сопряженных градиентов [36-38] использовались в данной работе для обучения нейронной сети.

5.1.3 Структурная схема алгоритма определения протокола информационного обмена

Структурная схема алгоритма классификации сетевых пакетов на принадлежность протоколам TLSv1 и TLSv1.2 представлена на рис.5., где входные показатели характеризуют:

Y_1 – номер порта [TCP/UDP] отправителя;

Y_2 – номер порта [TCP/UDP] получателя;

Y_3 – значение первого байта в последовательности payload протокола верхнего уровня;

Y_4 – степень принадлежности малому значению длины пакета (нечеткому множеству);

Y_5 – степень принадлежности большому значению длины пакета (нечеткому множеству);

Y_6 – степень принадлежности среднему значению длины пакета (нечеткому множеству);

Y_7 – степень принадлежности малому значению номера порта [TCP/UDP] отправителя (нечеткому множеству);

Y_8 – степень принадлежности большому значению номера порта [TCP/UDP] отправителя (нечеткому множеству);

Y_9 – степень принадлежности малому значению номера порта [TCP/UDP] получателя (нечеткому множеству);

Y_{10} – степень принадлежности большому значению номера порта [TCP/UDP] получателя (нечеткому множеству);

Y_{11} – принадлежность определенному в RFC 2246 и RFC 5246 интервалу целых чисел ContentType для протоколов TLSv1 и TLSv1.2 соответственно.

Выходные показатели имеют вид:

Z_1 – вероятность принадлежности к классу TLSv1 (результирующий показатель);

Z_2 – вероятность принадлежности к классу TLSv1.2 (результирующий показатель);

Z_3 – код внутреннего состояния наиболее вероятного класса (результирующий показатель).

Таким образом, поступающие на вход КСП признаки Y_4, Y_5, Y_6, Y_7, Y_8, Y_9 и Y_{10} являются нечеткими ЛП, прошедшими на этапе предобработки исходных данных процедуру фаззификации, которая заключается в том, что на вход блока обработки поступает последовательно сформированный массив IP-пакетов размерностью *W*. В массиве содержатся значения всех входных атрибутов . Целью этапа – получение значений функции принадлежности для всех условий из базы правил:

$$Y_n^w = \widetilde{X_n^w} = \mu\,(X_n^w) = \begin{cases} \mu(X_4^w) \\ \mu(X_5^w) \\ \cdots \\ \mu(X_{10}^w) \end{cases} \quad (15)$$

Таким образом, получается матрица множеств значений Y_n^w (или $\widetilde{X_n^w}$), где *w = 1, …, W* – анализируемые пакеты; *n =4,…,10* – количество исследуемых атрибутов пакета.

На первой ступени расчетов используются нечеткие контроллеры (НК) и модели на логических правилах (МЛП). Например, в блоке проверки несоответствия используемых номеров портов требованиям RFC на основе логических правил проверяются номера портов для исследуемых протоколов TLSv1 и TLSv1.2. Как правило, для передающей и принимающей сторон используется порт 443 и номер порта из интервала целых чисел, нижняя граница которого превышает 50000.А в блоке идентификации внутреннего состояния сетевого протокола для определения внутренние состояния используется hex-последовательность (из *Payload_hex*). Посредством логических правил происходит отождествление входных *hex*-значений с эталонными. При этом логическое правило для определения некоторого *k*-го внутреннего состояния некоторого *i*-го протоколаw-го пакета может иметь вид:

$$R\text{: ЕСЛИ } (H_i^w[0] == 0xA3) \text{ И } (H_i^w[1] == 0x3D) \text{ И } \cdots \text{ И } (H_i^w[n] == 0xC2) \text{ ТО } (SOST_i^w = S_k) \quad (16)$$

Вторая ступень использует методы логистической регрессии или алгоритмы нечеткой логики, а именно, алгоритм нечеткого вывода Мамдани. Результаты расчетов коэффициентов парной корреляции (таблица 2) входных признаков модели второй ступени показали, что входные признаки являются независимыми и могут эффективно использоваться в модели на основе логистической регрессии. Для подобных расчетов, как правило, используются различные математические пакеты. В данной работе применялся IBM SPSS Statistics 19 [39].

Наряду с независимостью входных признаков, интерес представляет значимость входных синтетических признаков, поступающих на вторую ступень расчетов. Используя количество ошибок классификации, если для классификации используется каждый из входных признаков в отдельности, а также метод анализа иерархий [40] и наши знания о сетевых протоколах прикладного уровня, мы определили весовые коэффициенты для признаков модели второй ступени по степени значимости для протоколов TLSv1 и TLSv1.2:

$$W_{ports} = 0{,}1267;\ W_{x_{11}} = 0{,}566;\ W_{length} = 0{,}0398;\ W_{sost} = 0{,}2674.$$

Таблица 2

Результаты расчетов коэффициентов парной корреляции

	PORTS	X11	LENGTH	SOST
PORTS	1	0,211	0,11	0,073
X11	0,211	1	0,397	0,176
LENGTH	0,11	0,397	1	0,188
SOST	0,073	0,176	0,188	1

Наиболее значимым признаком модели второй ступени расчетов является принадлежность определенному в RFC интервалу целых чисел ContentType для исследуемых протоколов TLSv1 и TLSv1.2. На втором месте по значимости определяемое в МЛП1 внутреннее состояние протокола. Следует также отметить, высокую полезность итогового внутреннего состояния исследуемых протоколов – Z_3. Данный результирующий признак очень полезен для дальнейшей оптимизации построенного классификатора сетевых пакетов прикладного уровня, о чем свидетельствуют результаты проведенных тестов.

На основе описанного выше двухступенчатого алгоритма определения протокола информационного обмена был разработан тестовый программный модуль.

6. Тестирование построенного классификатора сетевых пакетов

На данном этапе производятся расчеты с использованием искусственных нейронных сетей (ИНС) и алгоритмов нечеткой логики

Для использования в межсетевых экранах, в DPI-системах, в СОВ и в других, для операционных систем LINUX и операционных систем линейки Windows на языке программирования С++ построен программный модуль.

В таблице 3 приводятся характеристики используемой при тестировании аппаратной платформы.

Результаты тестирования представлены в таблицах 4 и 5.

В таблице 6 представлены измерения вычислительных ресурсов используемых вариантов моделей на второй ступени расчетов.

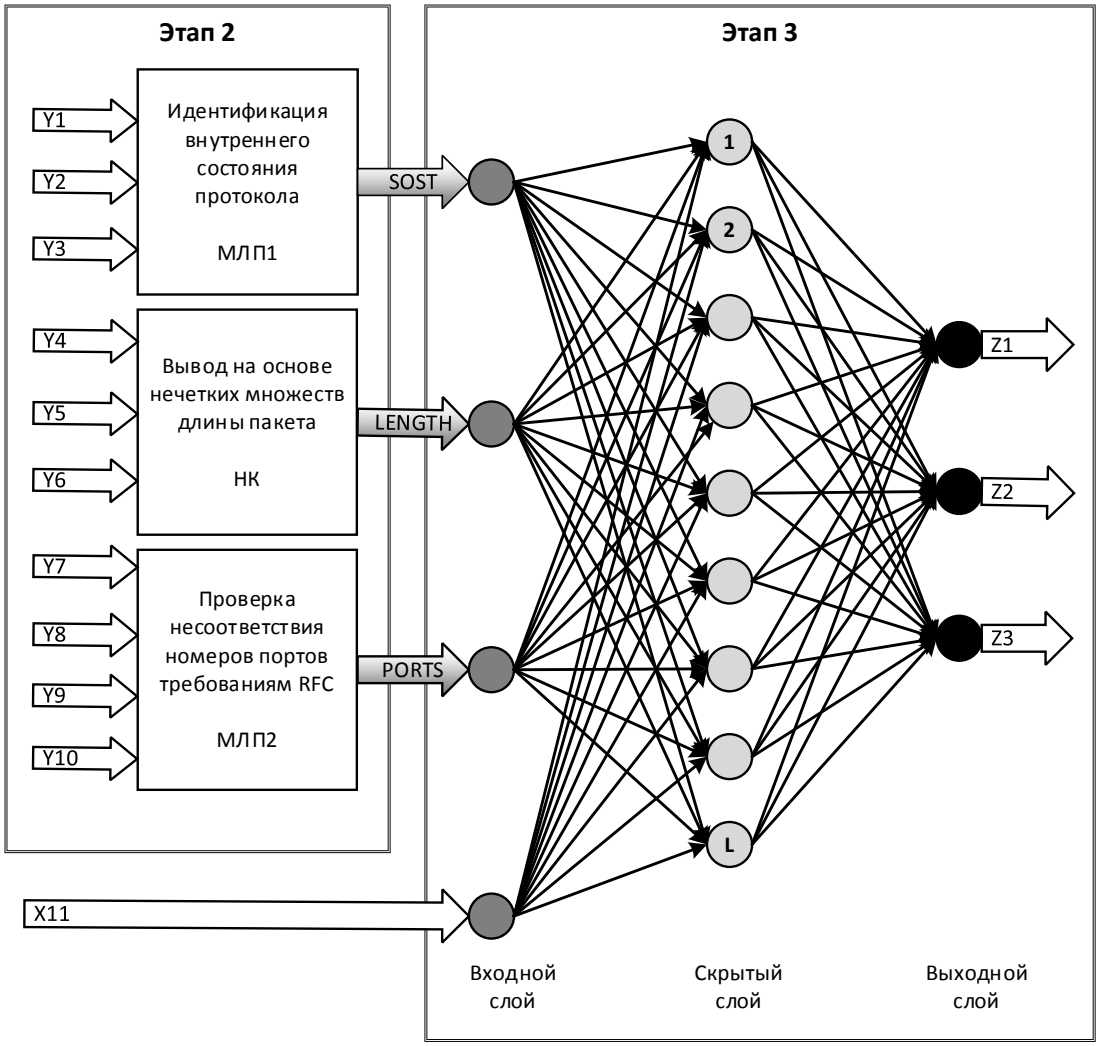

Рис. 5. Структурная схема двухступенчатого алгоритма классификации сетевых пакетов

Таблица 3

Характеристики используемой аппаратной платформы

Ресурсы	АП
Центральныйпроцессор	Intel Core i5-6400 2,7 GHz
Оперативнаяпамять	8 Gb
Установленное ПО	ОС MS Windows 10 Pro 64 bit

Основные преимущества представленного решения:

– высокая вычислительная производительность классификации с использованием нейронных сетей и алгоритмов нечеткой логики;

– возможность применения параллельных вычислений;

– высокая верность классификации и обобщение на тестовых выборках;

– совместимость с доверенными аппаратными платформами, такими как Эльбрус, и переносимость на различные операционные системы.

7. Заключение

Представленная в данной работе методика определения сетевых протоколов информационного обмена иллюстрирует современный подход по применению методов машинного обучения и алгоритмов нечеткой логики в интересах создания систем обеспечения информационной безопасности (IDS, IPS, NMS, DDoSPS и т.д.). Этот подход существенно отличается от применяемых в настоящее время алгоритмов классификации, основанных на анализе последовательности правил, предварительно подготовленных высококва-

Таблица 4

Результаты тестирования программного модуля для протоколов TLSv1 и TLSv1.2

Выборка Логистическаярегрессия		Не TLSv1 и не TLSv1.2		TLS v1		TLS v1.2	
		Логистическаярегрессия	С использованиемнечеткихмножеств	Логистическаярегрессия	С использованиемнечеткихмножеств	Логистическаярегрессия	С использованиемнечеткихмножеств
Тестовая выборка	Кол-вопакетов	881	881	104	104	269	269
	Ложноесрабатываение	5	5	75	17	10	18
	Верность	99%	99%	**28%**	*84%*	96%	93%

Таблица 5

Общие результаты тестирования программного модуля

Выборка Логистическаярегрессия		Не TLSv1 и не TLSv1.2 и не HTTP		HTTP		Общийрезультаттестирования	
		Логистическаярегрессия	С использованиемнечеткихмножеств	Логистическаярегрессия	С использованиемнечеткихмножеств	Логистическаярегрессия	С использованиемнечеткихмножеств
Тестовая выборка	Кол-вопакетов	881	881	25	25	1254	1254
	Ложноесрабатываение	5	5	1	2	91	42
	Верность	99%	99%	96%	92%	93%	97%

Таблица 6

Результаты вычислительной производительности

Метод	Логистическаярегрессия	С использованиемнечеткихмножеств
Среднее время классификации пакета, мс	0,755	0,6735

лифицированными специалистами в области информационной безопасности. Полученные результаты демонстрируют высокую верность классификации и вычислительную производительность классификации.

Предложен новый эффективный алгоритм

определения протоколов информационного обмена на основе классификации сетевых пакетов с использованием алгоритмов нечеткой логики и методов машинного обучения. Основной эффект алгоритма заключается в том, что для определения протокола используется принцип высокоскоростной однопакетной классификации, который позволяет анализировать информацию, передаваемую в каждом конкретном пакете. Используются элементы поведенческого анализа, а именно, классифицируются переходные состояния протоколов информационного обмена, что позволяет достичь более высокого уровня верности классификации и более высокой степени обобщения на новых тестовых выборках.

Дальнейшее развитие КСП связано с поиском новых способов определения DDoS-атак. На первом этапе с использованием КСП могут определяться протоколы информационного обмена прикладного уровня и типы устройств, задействованные при обмене. Далее могут быть выявлены приложения на устройствах информационного обмена и затем сетевые приложения прикладного уровня. Такой способ определения потенциальных сетевых угроз не является высокоскоростным. Однако он может быть очень эффективным в рамках тестовой среды для проведения полноценных комплексных исследований в задачах идентификации DDoS-атак.

Литература

1. Лось А.Б., Даниелян Ю.Ю. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ПРЕДСТАВЛЕННЫХ НА ОТЕЧЕСТВЕННОМ РЫНКЕ. Вестник МФЮА /2014. – №3. С.181 – 187.

2. Kawai H., Ata S., Nakamura N., Oka I.Identification of Communication Devices from Analysis of Traffic Patterns. Electric Industry Co., Ltd. Japan, 2018.

3. Агеев С.А., Саенко И.Б., Котенко И.В. Метод и алгоритмы обнаружения аномалий в трафике мультисервисных сетей связи, основанные на нечетком логическом выводе // Информационно-управляющие системы. 2018. №3. С. 61-68. Doi: 10.15217/issn1684-8853.2018.3.61.

4. Хазов В. 2016.Введение в DPI: Аналитика, обстановка на рынке и тренды. – URL: https://vasexperts.ru/blog/privet-mir/

5. Рыжков Д.О. ОПРЕДЕЛЕНИЕ ПРОТОКОЛА ПРИКЛАДНОГО УРОВНЯ ДЛЯ АНАЛИЗА СЕТЕВОГО ТРАФИКА С ПРИМЕНЕНИЕМ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ // Материалы IX Международной студенческой научной конференции «Студенческий научный форум». URL: https://scienceforum.ru/2017/article/2017032799.

6. Rehak M., Pechoucek M., Grill M., Stiborek J., Bartos K., and Celeda P (2009). Adaptive multiagent system for network traffic monitoring. IEEE Intelligent Systems. 2009. Vol. 24(3). Pp 16-25.

7. AnuGowsalya R.S., Miruna Joe Amali S. - SVM Based Network Traffic Classification Using Correlation Information // International Journal of Research in Electronics and Communication Technology. April - June 2014) Vol. 1.

8. Елагин В.С., Зарубин А.А., Онуфриенко А. В. Эффективность DPI-системы для идентификации трафика и обеспечения качества обслуживания OTT-сервисов // Наукоемкие технологии в космических исследованиях Земли. 2018. Т. 10. № 3. С.40-53. doi: 10.24411/2409-5419-2018-10074.

9. Singh J., M.J. NeneA Survey on Machine Learning Techniques for Intrusion Detection Systems // International Journal of Advanced Research in Computer and Communication Engineering. Vol.2, Issue 11, November 2013. Department of Computer Engineering, DIAT, Pune, India. Pp 4349 – 4355.

10. Abraham S. and Nair S. Cyber Security Analytics: A stochastic model for Security Quantification using Absorbing Markov Chains // Journal of Communications. December 2014. Vol. 9(12): pp. 899-907.

11. Multi-level Machine Learning Traffic Classification System.Szabo G., Szule J., Turanyi Z., Pongracz G. // ICN 2012: The Eleventh International Conference on Networks. Pp 69 – 77.

12. Traffic Classification Using Probabilistic Neural Networks.Sun R., Yang B., Peng L., Chen Z., Zhang L., and Jing S. // Sixth International Conference on Natural Computation (ICNC 2010). Pp. 1914-1919.

13. Бабенко Г.В. Анализ современных угроз информации, возникающих при сетевом взаимодействии. Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика. 2010 №2.– URL:http://www.cosmos.ru/earth/trudi/1-28.pdf.

14. Sanders C. Practical packet analysis: 2nd edition. 2011. No Starch Press, Inc. 38 Ringold Street, San Francisco, CA 94103.

15. Internet Traffic Classification Demystified: On the Sources of the Discriminative Power.Lim Y., Kim H., Jeong J., Kim C., Kwon T., Choi Y. 2010. – URL: http://conferences.sigcomm.org/co-next/2010/CoNEXT_papers/09-Lim.pdf

16. Izenman A.J.Linear Discriminant Analysis // In: Modern Multivariate Statistical Techniques. Springer Texts in Statistics. Springer, New York, NY. 2013.

17. Bourez C.Deep learning with Theano. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 284 p.

18. Mamdani E.H., Assilian S.An experiment in linguistic synthesis thesis with a fuzzy logic controller.// International Journal of Man-Machine Studies, 1975. Vol. 7, no. 1, pp. 1 – 13.

19. Mamdani E.H. Advances in the linguistic synthesis of fuzzy controllers// International Journal of Man-Machine Studies, 1976. Vol. 8, pp. 669 – 678.

20. Mamdani E.H. Applications of fuzzy logic to approximate reasoning using linguistic synthesis // IEEE Transactions on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.

21. Пегат А.Нечеткое моделирование и управление / пер. с англ. – М. : БИНОМ. Лаборатория знаний, 2009. – 312 с.

22. Rumelhart D.E., Hinton G.E., Williams R.J.Learning Representations by Backpropagating Error // LETTERS TO NATURE. Vol. 323, 1986. Pp 533 – 536.

23. Bishop C.M.Neural Networks for Pattern Recognition // Department of Computer Science and Applied Mathematics Aston University Birmingham, UK. 1995. P. 479.

24. Рашка С.Python и машинное обучение / пер. с англ. А.В. Логунова. – М.: ДМК Пресс, 2017. – 418 с.

25. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / пер. с англ. А.А. Слинкина. – М.: ДМК Пресс, 2015. – 400 с.

26. Garreta R., Moncecchi G. Learning scikit-learn: Machine Learning in Python. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2013. 99 p.

27. Gulli A., Pal S.Deep Learning with Keras // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 318 p.

28. Изучаем Spark: молниеносный анализ данных.Карау Х., Конвински Э., Венделл П., Захария М. – М. : ДМК Пресс, 2015. – 304 с.

29. Pentreath N. Machine Learning with Spark. Packt PublishingLtd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2015. – 304 с.

30. McClure N.TensorFlow Machine Learning Cookbook // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017. 422 p.

31. Abrahams S., Hafner D., Erwitt E., Scarpinelli A.Tensorflow for machine intelligence. Bleeding Edge Press, Santa Rosa, CA 95404. 2016. 298 p.

32. Bonnin R.Building Machine Learning Projects with TensorFlow // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2016. 282 p.

33. Kaehler A., Bradski G.Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, 1st O'Reilly Media, Inc. ©2016 .

34. Beyeler M.Machine Learning for OpenCV // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 382 p.

35. Бычков И.Н., Глухов В.И., Трушкин К.А.Доверенная программно-аппаратная платформа «Эльбрус». Отечественное решение для АСУ ТП КВО // ИСУП – №1 (49).

36. Nocedal J., Wright S.J.Numerical Optimization ,Springer, 1999. 663 p.

37. Васильев Ф. П.Методы оптимизации - Издательство «Факториал Пресс», 2002. 824 с.

38. Галушкин А.И.Нейронные сети: основы теории. 2012. Горячая Линия – Телеком. 496 с.

39. Наследов А. SPSS 19: профессиональный статистический анализ данных. – СПб.: Питер, 2011. – 400 с.

40. Саати Т.Л. Взаимодействие в иерархических системах // Техническая кибернетика. 1979. №1. с. 68-84.

References

1. Los A. B., DanielyanYu.Yu. COMPARATIVE ANALYSIS OF THE SYSTEMS OF DETECTION OF INTRODUCTION REPRESENTED IN THE DOMESTIC MARKET. Vestnik MFLA No.3 / 2014, pp 181-186.

2. Kawai H., Ata S., Nakamura N., Oka I. Identification of Communication Devices from Analysis of Traffic Patterns. ElectricIndustryCo., Ltd. Japan, 2018.

3. Ageev S.A., Saenko I.B., Kotenko I.V. Method and algorithms for detecting anomalies in the traffic of multiservice communication networks based on fuzzy inference // Information-control systems.. 2018. №3. P. 61-68. Doi: 10.15217/issn1684-8853.2018.3.61.

4. Khazov V. 2016. Introduction to DPI: Analytics, market conditions and trends. – URL: https://vasexperts.ru/blog/privet-mir/

5. Ryzhkov D.O. DETERMINING THE APPLICATION LEVEL PROTOCOL FOR ANALYSIS OF NETWORK TRAFFIC USING MACHINE LEARNING ALGORITHMS. 9th International Student Scientific Conference: Student Scientific Forum - 2017. International Student Scientific Bulletin.URL: https://scienceforum.ru/2017/article/2017032799.

6. Rehak M., Pechoucek M., Grill M., Stiborek J., Bartos K., and Celeda P (2009). Adaptive multiagent system for network traffic monitoring. IEEE Intelligent Systems. 2009. Vol. 24(3). Pp 16-25.

7. AnuGowsalya R.S., Miruna Joe Amali S. - SVM Based Network Traffic Classification Using Correlation Information // International Journal of Research in Electronics and Communication Technology. April - June 2014) Vol. 1.

8. Elagin V.S., Zarubin A.A., Onufrienko A.V. The effectiveness of a DPI system for identifying traffic and ensuring the quality of service for OTT services // High-tech in space exploration of the Earth. 2018. T. 10. No. 3. Pp. 40-53. doi: 10.24411/2409-5419-2018-10074.

9. Singh J., M.J. Nene A Survey on Machine Learning Techniques for Intrusion Detection Systems // International Journal of Advanced Research in Computer and Communication Engineering. Vol.2, Issue 11, November 2013. Department of Computer Engineering, DIAT, Pune, India. Pp 4349 – 4355.

10. Abraham S. and Nair S. Cyber Security Analytics: A stochastic model for Security Quantification using Absorbing Markov Chains // Journal of Communications. December 2014. Vol. 9(12): pp. 899-907.

11. Multi-level Machine Learning Traffic Classification System. Szabo G., Szule J., Turanyi Z., Pongracz G. // ICN 2012: The Eleventh International Conference on Networks. Pp 69 – 77.

12. Traffic Classification Using Probabilistic Neural Networks. Sun R., Yang B., Peng L., Chen Z., Zhang L., and Jing S. // Sixth International Conference on Natural Computation (ICNC 2010). Pp. 1914-1919.

13. Бабенко Г.В. Анализ современных угроз информации, возникающих при сетевом взаимодействии. Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика. 2010 №2. – URL: http://www.cosmos.ru/earth/trudi/1-28.pdf.

14. Sanders C. Practical packet analysis: 2nd edition. 2011. No Starch Press, Inc. 38 Ringold Street, San Francisco, CA 94103.

15. Internet Traffic Classification Demystified: On the Sources of the Discriminative Power. Lim Y., Kim H., Jeong J., Kim C., Kwon T., Choi Y. 2010. – URL: http://conferences.sigcomm.org/co-next/2010/CoNEXT_papers/09-Lim.pdf

16. Izenman A.J. Linear Discriminant Analysis // In: Modern Multivariate Statistical Techniques. Springer Texts in Statistics. Springer, New York, NY. 2013.

17. Bourez C. Deep learning with Theano. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 284 p.

18. Mamdani E.H., Assilian S. An experiment in linguistic synthesis thesis with a fuzzy logic controller.// International Journal of Man-Machine Studies, 1975. Vol. 7, no. 1, pp. 1 – 13.

19. Mamdani E.H. Advances in the linguistic synthesis of fuzzy controllers// International Journal of Man-Machine Studies, 1976. Vol. 8, pp. 669 – 678.

20. Mamdani E.H. Applications of fuzzy logic to approximate reasoning using linguistic synthesis // IEEE Transactions on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.

21. Pegat A. Fuzzy modeling and control / A. Pegat; per. from English - M.: BINOM. Laboratory of Knowledge, 2009. – 312 p.

22. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Representations by Backpropagating Error // LETTERS TO NATURE. Vol. 323, 1986. Pp 533 – 536.

23. Bishop C.M. Neural Networks for Pattern Recognition // Department of Computer Science and Applied Mathematics Aston University Birmingham, UK. 1995. P. 479.

24. Raska S. Python and machine learning / Per. from English A.V. Logunova. - M.: DMK Press, 2017. – 418 p.

25. Flach P. Machine Learning. Science and the art of constructing algorithms that extract knowledge from data. from English A.A. Slinkin. - M.: DMK Press, 2015 .- 400 p.

26. Garreta R., Moncecchi G. Learning scikit-learn: Machine Learning in Python. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2013. 99 p.

27. Gulli A., Pal S. Deep Learning with Keras // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 318 p.

28. Karau H., Konvinsky E., Wendell P., Zachariah M. We study Spark: lightning-fast data analysis. - M .: DMK Press, 2015. – 304 p.

29. N. Pentreath. Machine Learning with Spark. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2015. – 304 c.

30. McClure N. TensorFlow Machine Learning Cookbook // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017. 422 p.

31. Abrahams S., Hafner D., Erwitt E., Scarpinelli A. Tensorflow for machine intelligence. Bleeding Edge Press, Santa Rosa, CA 95404. 2016. 298 p.

32. Bonnin R. Building Machine Learning Projects with TensorFlow // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2016. 282 p.

33. Kaehler A., Bradski G. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, 1st O'Reilly Media, Inc. ©2016 .

34. Beyeler M. Machine Learning for OpenCV // Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2017. 382 p.

35. Bychkov I.N., Glukhov V.I., K.A. Trushkin. 2014 Trusted software and hardware platform "Elbrus". Domestic solution for ACS TP KVO. ISUP Magazine No. 1 (49).

36. NocedalJ., WrightS.J. NumericalOptimization ,Springer, 1999. 663 p.

37. Vasiliev F. P. Optimization Methods - Factorial Press Publishing House, 2002, 824 p.

38. Galushkin A.I. 2012. Neural networks: the basics of theory. Hot Line - Telecom. ISBN 978-5-9912-0082-0, 496 p.

39. Nasledov A. SPSS 19: professional statistical data analysis. - St. Petersburg: Peter, 2011 .-400 p.

40. Saati T.L. Interaction in hierarchical systems // Technical cybernetics. 1979. No. 1.with. Pp 68-84.

ЕРМАКОВ Роман Николаевич, кандидат биологических наук, ведущий инженер Научно-исследовательский институт «Масштаб». 194100, г. Санкт-Петербург, ул. Кантимировская, д. 5. E-mail: romul151925@mail.ru

ERMAKOV Roman Nikolaevich, Candidate of Biological Sciences, Leading Engineer of the Scientific-Research Institute "Masshtab". 194100, St. Petersburg, Kantimirovskaya, д. 5. E-mail: romul151925@mail.ru