computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

NETWORK APPLICATION-LAYER PROTOCOL CLASSIFICATION BASED ON FUZZY DATA AND NEURAL NETWORK PROCESSING

Vyacheslav Efimov [1)], Igor Kotenko [2)], Igor Saenko [2)]

[1)] Joint Stock Company "Research Institute "Masshtab"; 5A, Kantemirovskaya St., 194100, St. Petersburg, Russia,
v.efimov@mashtab.org
[2)] Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Sciences,
ivkote@comsec.spb.ru, ibsaen@comsec.spb.ru

Paper history:
Received 03 June 2020
Received in revised form 31 August 2020
Accepted 2 September 2020
Available online 27 September 2020

Keywords:
classification of network packets;
artificial intelligence;
artificial neural networks;
fuzzy logic;
network traffic analysis;
deep package analysis;
machine learning.

Abstract: A technique of network packet classification on the application layer is proposed. It is based on fuzzy data processing and artificial neural networks to define the network packet belongingness to one of the known network protocols. In the suggested technique, two main data processing stages are distinguished. At the first stage data is preprocessed by fuzzy logic methods. At the second stage the packets are classified by means of an artificial neural network. An artificial neural network having the proposed architecture allows one to determine the following aspects: the type of secure network protocol, the internal state of the network protocol based on the application of logical decision rules, and the type of network application using the identified protocol. The architecture of the bench environment for field tests is considered. During the experiments, the traffic of real network applications that are used around the world was used. Experimental assessment of the offered technique showed rather high quality and work speed of the developed classifier.

1. INTRODUCTION

The current stage of development of almost all sectors of the economy, including energy, manufacturing, finance, etc., is characterized by their wide informatization, which is based on the intensive development of communication infrastructure and the massive use of information technology. However, the infinity of the global information and communication space leads to the possibility of intercepting information by different malefactors and the unlawful use of information technology. First of all, this is due to the possibility of carrying out computer attacks against the information resources of corporations and enterprises. In this case, telecom operators and information system developers are forced to pay increased attention to issues related to the construction of security systems. One of the most important issues of this kind is the preservation of confidential information in compliance with international standards and protocols of information and communication interaction.

The indicated security systems are built on the basis of the following systems:

- Identity Management (IDM) systems;
- Privilege Accounts Management (PAM) systems;
- Next Generation Firewalls (NGFW);
- Security Information and Event Management (SIEM) systems;
- Antivirus (AV), Antibot, Malware Protection systems;
- Intrusion Detection Systems (IDS), including Application protocol-based IDS (APIDS);
- Intrusion Prevention Systems (IPS)
- Network Traffic Analyzers (NTA);
- DDoS Protection Systems (DDoS PS);
- Policy and Charging Enforcement Function (PCEF) systems;
- Policy and Charging Rules Function (PCRF) systems;
- Network Access Control (NAC) systems;
- and other systems.

Information security systems can and should be combined and be information providers for other systems of this type. For example, NTA systems can provide information on security incidents for IDS, IPS, NMS, DDoS PS, and others. Reliable classification of Internet protocols will provide the necessary information to the above systems to identify:

- network attacks, abnormal and (or) fake traffic [1];
- network devices [2] ;
- application-level software operating at the seventh layer of the OSI model (Skype, Facebook, Viber, Telegram and others) [3, 4] and their states.

The classification of network protocols is implemented by the following methods:

- signature, behavioral and hybrid analysis of network packets. The signature method is based on the analysis of the header and the information block (payload) of packets. The behavioral method is based on the study of the statistical properties of network traffic: the sequence, number and size of network packets, time intervals between packets, etc. [5–7];
- artificial intelligence (machine learning algorithms [8], neural networks [9], fuzzy logic, genetic algorithms [10], etc.);
- traditional mathematical research (fractal and wavelet analysis [4], cluster analysis [11], etc.);
- combination and development of various classification methods.

Methods for traffic classification in the interests of solving the problem of detecting network attacks are presented in [12-14]. This list of publications is not exhaustive.

The indicators of the classification efficiency of network protocols are as follows: classification accuracy, computational complexity of the classification algorithm, the ability of the algorithm to parallelize calculations, classification time, and others. Increasing the values of these indicators has an important impact on the functioning of NTA, DPI [15, 16], IDS / IPS [17], DDoS PS and other systems.

The main method of traffic classification in these systems is based on the signature based approach, which is characterized by high resource consumption. A qualified estimate of the complexity of the signature based algorithms is given in [12]. In addition, it can be clarified that these classical methods have a multitude of disadvantages:

- the number of false positives is potentially higher due to the use of encryption mechanisms;
- the accuracy of the network packet classification depends on the competence of system administrators in configuring the system;
- do not allow classification of protocols having a variable part of packet attributes (variable port, IP address, encryption);
- it is not possible to compile a signature for each protocol.

The following approaches are distinguished by the level of classification of network packets:

- Shallow Packet Inspection (SPI);
- Medium Packet Inspection (MPI);
- Deep Packet Inspection (DPI).

The analyzers of the "shallow" level function in the simplest firewalls where the decision on blocking of packets is usually made according to the list of the prohibited IP addresses and port numbers.

The analyzers of "medium" level allow one to carry out traffic filtering by using the information on the transmitted data format and also on more complete localization of the sender. These tools usually act as the intermediary (proxy) between an access provider to the Internet and an internal network.

The systems of "deep" packet analysis are intended to identify the applications participating in network interactions and define the states of information exchange protocols. Therefore the "deep" analysis of network packets assumes the analysis of content of these packets on all levels. The purpose of the DPI systems is ensuring the control over execution of the requirements for information security of info-telecommunication infrastructure and monitoring the quality of functioning of communication channels.

The paper suggests a technique of network packet classification on the application layer which can be used both on the level of Medium Packet Inspection and Deep Packet Inspection (DPI). The novelty of the paper is in a new approach for combining fuzzy data processing and artificial neural networks to define the belongingness of network packets to one of the known network protocols. Thus, the paper suggests using two stages in the offered technique: at the first stage data is preprocessed by fuzzy logic methods. At the second stage, the packets are classified by means of an artificial neural network. Experimental assessment of the offered technique showed rather high quality and work speed of the developed classifier.

Statement of the network protocol classification problem is considered in the second section. The third section suggests the general description of the traffic classification technique used. The fourth section specifies the preprocessing stage of the suggested approach. The technique for neural network processing is considered in the fifth section. The sixth section presents the network packet classifier implementation and the results of experiments. The seventh section summarizes the main results and reveals the direction of further research.

2. STATEMENT OF THE NETWORK PROTOCOL CLASSIFICATION PROBLEM

The problem of the network protocol classification can be formulated as follows.

There is a set of the investigated objects - IP packets of the application layer:

$$P = \{P_1, P_2, \cdots, P_w, \dots, P_W\}\,, \tag{1}$$

where P_w is an analyzed packet from the sequence of packets (traffic) with dimension from 1 to W.

Each object (IP packet) is characterized by a set of variables (attributes):

$$P_w = \{X_1^w, X_2^w, \cdots, X_i^w, \cdots, X_{13}^w, P, Z\}\,, \tag{2}$$

where X_i^w is i-th observable attribute of w-th packet, which value is defined in "Request for comments" (RFC) of the known classified protocol, $i = 1, \dots, 13$; P is a useful packet data (payload); Z is a dependent set of feature values defined at classification of the protocol.

The set Z includes:

$$Z = \{Z_t, Z_s, Z_p\}, \tag{3}$$

where Z_t is a type of the protocol; Z_s is a state of the protocol; Z_p is a type of the application.

Each attribute X_n takes a value from some set:

$$\mathbf{X}_n = \{Xn_1, Xn_2, \cdots, Xn_i, \cdots, Xn_M\}, \tag{4}$$

where Xn_i is the i-th variant of attribute values from M possible variants described in RFC.

Thus, the problem of classification comes down to definition of the set Z based on the values of attributes of the packet sequence Xn_i.

Solving an applied classification problem, taking into account the analysis of researches [14, 18], the following set of important attributes of packets (factor space) was defined:

- X_1 - *EtherType* (a type of the Ethernet protocol standard);
- X_2 - *Multicast* (it is set to 1 if multicast, otherwise – 0);
- X_3 - *IPprotocol* (a type of the transport level);
- X_4 - *Packet Length* (length of a network packet in bytes);
- X_5 - *Source IP Address* (IP address of a sender);
- X_6 - *Destination IP Address* (IP address of a recipient);
- X_7 - *Source Port* (a sender TCP/UDP port);
- X_8 - *Destination Port* (a destination TCP/UDP port);
- X_9 - *Serial Number of the Packet;*
- X_{10} - *The Confirmation Number of the Packet*;
- X_{11} - *Markers* (used to maintain the quality of the service and indicate priority when processing the package);
- X_{12} - *Length* (number of bytes of the payload hexadecimal set);
- X_{13} – *Teaching* (the protocol which value is a priori known during the training, null – in other cases);
- $P = \{P_1,\ P_2, \dots, P_J\}$ - a hexadecimal Payload set with dimension of J bytes (IP packet payload).

Network packets may be as shown in Table 1 (for attributes *Ether Type, Packet length, Source IP Address,* and *Destination IP Address*), Table 2 (for attributes *Source Port, Destination Port, Serial Number,* and *Confirmation Number*) and Table 3 (for attributes *Markers, Length, Payload,* and *Teaching*).

Table 1. The variant of the packet sequence in the network traffic

N	Ether Type	Packet length	Source IP Address	Destination IP Address
0	0x800	73	172.16.0.1	172.16.0.10
1	0x800	72	172.16.0.1	172.16.0.10
2	0x800	90	172.16.0.1	172.16.0.10
3	0x800	91	192.168.10.3	192.168.10.8
4	0x800	459	13.79.241.1	192.168.10.1
5	0x800	91	192.168.10.3	192.168.10.8
6	0x800	491	94.100.181.5	192.168.10.5
7	0x800	199	192.168.10.3	192.168.10.8
8	0x800	113	192.168.10.3	192.168.10.8
9	0x800	267	192.168.10.1	13.79.241.1
10	0x800	571	13.79.241.1	192.168.10.1
11	0x800	731	13.79.241.1	192.168.10.1
12	0x800	305	192.168.10.7	40.115.1.4
13	0x800	1223	40.115.1.4	192.168.10.7
14	0x800	268	192.168.10.7	40.115.1.4

Table 2. The variant of the packet sequence in the network traffic

N	Source Port	Destination Port	Serial Number	Confirmation Number
0	53986	21	0x0001	0x0000
1	53986	21	0x0002	0x0000
2	53986	21	0x0003	0x0000
3	443	61983	0x0002	0x0000
4	443	61867	0x0004	0x0001
5	443	61986	0x0003	0x0000
6	443	61662	0x0012	0x0000
7	443	61993	0x0003	0x0000
8	62544	443	0x0004	0x0000
9	61988	443	0x0010	0x0000
10	443	61990	0x0005	0x0000
11	443	61988	0x0006	0x0000
12	61991	443	0x0001	0x0000
13	443	61991	0x00FF	0x0000
14	61991	443	0x0002	0x0000

Table 3. The variant of the packet sequence in the network traffic

N	Markers	Length	Payload	Teaching
0	0xF1	7	504153	FTP
1	0xF1	6	4c4953	FTP
2	0xF1	24	504f52	FTP
3	0x0A	37	150301	TLSv1.1
4	0x00	405	170303	TLSv1.1
5	0x0A	37	150301	TLSv1.1
6	0x00	437	170301	TLSv1.1
7	0x40	145	170303	TLSv1.1
8	0x40	59	170303	TLSv1.1
9	0x00	213	d2a6ea	TLSv1.2
10	0x00	517	6edb93	TLSv1.2
11	0x00	677	170303	TLSv1.2
12	0x00	251	170303	TLSv1.2
13	0x00	1169	05bb0f	TLSv1.2
14	0x00	214	170303	TLSv1.2

3. GENERAL DESCRIPTION OF THE TRAFFIC CLASSIFICATION TECHNIQUE

When implementing "deep" packet analysis, the present paper considers a combined method of traffic classification based on the application of neural networks and fuzzy sets [19-24]. Significant gain in calculation time while solving traffic classification problem is achieved due to reduction of factor space by introducing two-stage method of processing (Fig. 1), which includes two stages: pre-processing stage and neural network processing stage.

The use of fuzzy sets allows one to expand the understanding of ordinary mathematical sets. In this case, the binary nature of belongingness of some element to the set is rejected, i.e., the membership function takes the value 1 (“true”) when the element belongs to the set and the value 0 (“false”) when it does not belong [25]. The membership function of a fuzzy set can take any values on the interval [0, 1].

The advantage of artificial neural models is expressed in the ability to analyze incomplete input data or data with natural noise, or data obtained as a result of exposure to the system. Algorithms based on neural networks process each event that has its own weight, which is important for traffic analysis. Algorithms are implemented by elementary mathematical operations, due to which they have a high speed of operation. They have the possibility of self-learning and the ability to predict further events in the system. The indicated possibilities of mathematical methods allow one to suggest that their implementation in software will minimize the time of network traffic classification and increase the volume of the transmitted traffic, which is an urgent task in the condition of increased workload in information and communication networks.

Many authors have already applied the presented mathematical methods [21, 26-28] separately. It is proposed here to apply them together in order to reduce the dimension of the problem being solved and increase the efficiency of solving the IP packet classification problem. As a result, on the one hand, it will reduce the requirements for computational resources of NTA, IDS/IPS, DDoS PS and others systems and, on the other hand, will increase their efficiency.

In the first stage (pre-processing stage) the following operations are performed:

1. Primary determination whether the network packet being analyzed belongs to specific groups ($GROUP_A$, $GROUP_B$, $GROUP_C$)
2. Fuzzification and normalization of attribute values
3. Reduction of the dimension of factor space of features (convolution)
4. Defuzzification.

In the second step (neural network processing stage), using neural network processing, the traffic classification is completed using the method of logical regression. The result of this step is the calculation of the dependent set Z.

4. PREPROCESSING STAGE

When preprocessing the packets of the analyzed traffic, the following operations are performed:

1. Allocation of structured data of the IP packet,
2. Identification of the internal state of the protocol (Fig. 1, block A). Preparation for classification is provided by using additional features of internal state of the protocol (connection of subscribers, key exchange, identification, data transmission, session completion, etc.);
3. Division of the classified network packets of the relevant protocols (DHCP, DNS, FTP, NTP, HTTP, HTTPS, SSL, TLS, etc.) on homogeneous groups ($GROUP_A$, $GROUP_B$, $GROUP_C$) (Fig. 1, block B);
4. Normalization of attributes (Fig. 1, block B). Normalization is the process of bringing attribute values to the same scale;
5. Preliminary classification based on the application of fuzzy set theory and convolution (Fig. 1, block C). Calculations are transformations of normalized attributes based on logical rules and fuzzy logic algorithms. Actions are completed by convolution of normalized attributes to reduce the dimension of tasks.

Three groups of protocols are defined as follows:

- $GROUP_A$: the group of protocols in which connections of subscribers are established – TCP protocols;

- $GROUP_B$: the group of protocols in which connections of subscribers are not established – UDP protocols. Traffic is processed by two subscribers;
- $GROUP_C$: the group of protocols in which connections of subscribers are not established – UDP protocols. Traffic from one subscriber is processed at the same time by several subscribers.

Division is carried out on the basis of values of attributes (*EtherType*, *Multicast* и *IPprotocol*) using the database of the public information resource Internet Assigned Numbers Authority (IANA) [29].

4.1 IDENTIFICATION OF THE INTERNAL STATE OF THE PROTOCOL

For identification of a state of the protocol (Fig. 1, block A) it is not enough to use data from a packet header. It is necessary to retrieve additionally the data from the packet Payload field (attribute P). In this case the identifying features of the protocol status are the data retrieved from hexadecimal useful data of the transport layer packets of the Payload field. Internal state is defined on the basis of the logical decisive rules constructed on the basis of data of RFC (TLS 1.0 RFC 2246, TLS 1.1 RFC 4346, TLS 1.2 RFC 5246, TLS 1.3 RFC 8446) [30] where the logic of work of the protected transport layer protocol is specified. Main session states of exchange of TLS are initial connection, exchange of cryptographic keys, determination of connection parameters, authentication, warning, data exchange, and session completion.

For example, the rules of definition of connection state may be:

SessionTLS = -1;
IF (P[2] == 0x03) AND (P[3] == 0x00) then SessionTLS = 0;
IF (P[2] == 0x03) AND (P[3] == 0x01) then SessionTLS = 1;
IF (P[2] == 0x03) AND (P[3] == 0x02) then SessionTLS = 2;
IF (P[2] == 0x03) AND (P[3] == 0x03) then SessionTLS = 3.

Thus, the output of block A, in case of establishment of a protocol state Y1, will have positive value. The value at the Y1 output provides additional input data for the neural network. The neural network works more accurately using Y1.

4.2 DIVISION

The division of classified network packets P_w is based on the following logical rules:

$$\begin{gathered} GROUP_A = \emptyset; GROUP_B = \emptyset; GROUP_C = \emptyset \\ IF\ (X_2 == 0)\ \text{AND}\ (X_1 == \text{TCP})\ \text{AND} \\ (X_3 == 0\text{x}04)\ \text{THEN}\ P_w => GROUP_A; \\ IF\ (X_2 == 0)\text{AND}\ (X_5 == \text{UDP})\ \text{AND} \\ (X_3 == 0\text{x}04)\ \text{THEN}\ P_w => GROUP_B; \\ IF\ (X_2 == 1)\ \text{AND}\ (X_5 == \text{UDP})\ \text{AND} \\ (X_3 == 0\text{x}04)\ \text{THEN}\ P_w => GROUP_C. \end{gathered} \tag{5}$$

The distribution of network packets to these sets may be represented as follows:

$$\begin{gathered} GROUP_A = \left\{ \begin{matrix} \text{TCP}, \\ \text{TLSv1.0}, \\ \text{TLSv1.1}, \\ \text{TLSv1.2}, \\ \text{TLSv1.3}, \\ \text{SSHv2}, \\ \text{HTTP}, \\ \text{HTTPS} \end{matrix} \right\}; \\ GROUP_B = \left\{ \begin{matrix} \text{UDP}, \\ \text{STUN}, \\ \text{QUIC}, \\ \text{NBNS}, \\ \text{DNS}, \\ \text{BROWSER} \end{matrix} \right\}; \\ GROUP_\text{C} = \left\{ \begin{matrix} \text{SSDP}, \\ \text{MDNS}, \\ \text{LLMNR} \end{matrix} \right\}. \end{gathered} \tag{6}$$

4.3 FUZZIFICATION AND NORMALIZATION

The stage of a fuzzification and normalization of entrance attributes of a packet $\{X_1^w,...,X_{13}^w,P\}$ of the protocol is carried out with the use of member functions (μ). On an input of a processing unit, a consistently created array of IP packets dimension of W arrives. The array contains values of all input attributes X_n^w. The purpose of a stage is obtaining member function values for all conditions from the rule base:

$$\widetilde{X_n^w} = \mu\ (X_n^w) = \begin{cases} \mu(X_1^w) \\ \mu(X_2^w) \\ \cdots \\ \mu(X_{13}^w) \end{cases}. \tag{7}$$

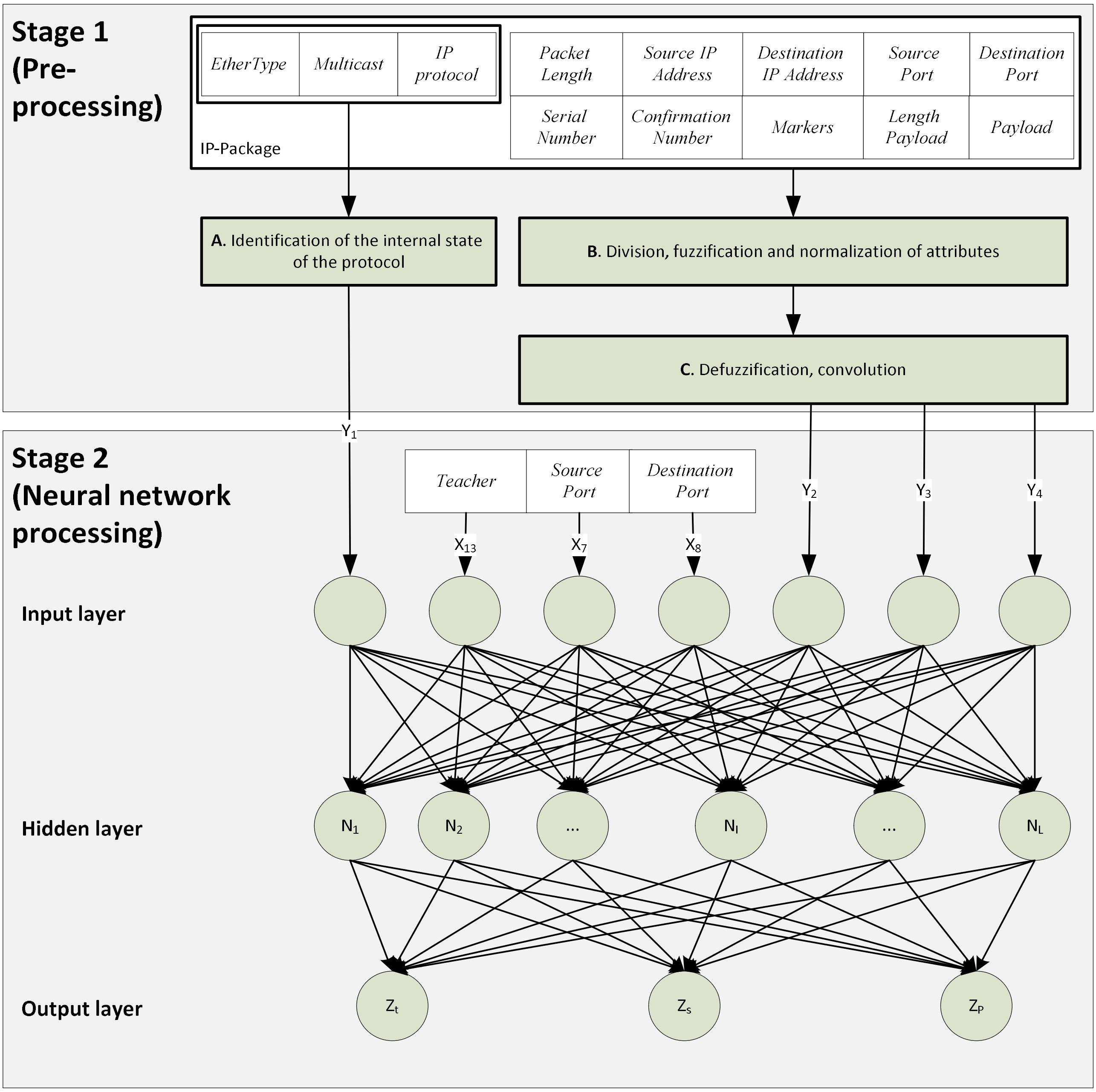

Figure 1 – Structural diagram of the two-stage network packet classification algorithm

Thus, the matrix $\widetilde{X_n^w}$ is a set of results of calculating the membership function for the n-th attributes of w-th IP packet, where w = 1, ..., W is the number of the classified packets; n =1,…, 13 is the number of packet attributes investigated.

We used linear membership functions (sigmoid, triangular, trapezoidal and other species) based on the rules, for example:

HTTPS_RULE: IF (X_7==443)
THEN ($\widetilde{X_7}$ = 1);
HTTPS_RULE: IF (X_7>=50000) (8)
THEN ($\widetilde{X_7}$ = 1)

HTTPS_RULE: IF (X_8<=443) *AND*
(X_8<=50000) (9)
THEN ($\widetilde{X_8}$ = 1)

In the presented rule set the values of port numbers of the TLS protocol, defined on the basis of data from RFC [30], are used. On the basis of simple logical processing rules of the port number X7, the studied protected protocols with enciphering TLSv1.0, TLSv1.1, TLSv1.2 and TLSv1.3 are classified. As a rule, for the transferring and accepting parties, the port 443 and ports from an interval of integer numbers, the lower bound value of which exceeds 50000, are used. The variant of graphical representation of the membership function constructed according to the rule base (8) for X7 is presented in Fig. 2-a. Fig. 2-b depicts the graphical representation of the membership function constructed according to the rule base (9) for X8. Generally different models of normalization functions of both linear, and not linear type can be applied. The influence of different types of membership functions on quality of classification will be the next stage of investigations.

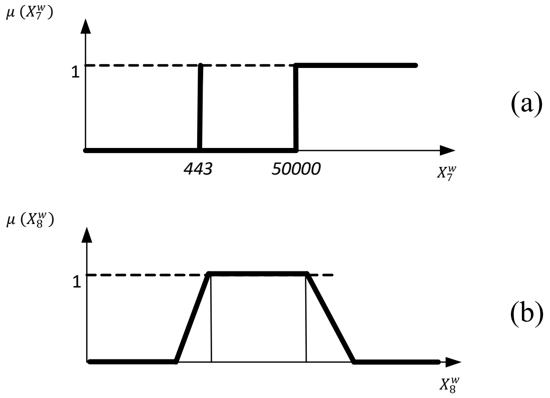

Figure 2 – Variants of graphical representation of membership functions: (a) Processing of the attribute *X7*; *(*b) Processing of the attribute *X8*

Results of calculations will be the values of the membership function within [0, 1]. Therefore, additional processing of values of the attributes X7 and X8 for normalization is not required. Normalization of values of other attributes of the IP packet is carried out by standard conversions of reduction to an identical scale in the range of [0…1] by the following expression:

$$\widetilde{X_n} = \frac{X_n - X_{n\_min}}{X_{n\_max} - X_{n\_min}}, \qquad (10)$$

where X_{n_min} is the minimal value of the attribute X_n; X_{n_max} is the maximal value of the attribute X_n.

4.4 DEFUZZIFICATION

The algorithm of defuzzification of output variables is carried out on the basis of the algorithm of Mamdani-Zade fuzzy inference [31-33], which purpose is obtaining quantitative value for each of output linguistic variables. Formally it occurs as follows: the output variable $\widetilde{X_\iota^w}$ and the set $\widetilde{X_\iota^w}$ (i = 1, …, 13) are considered, then a total quantitative value of each output variable is calculated. The value at the exit of the model is calculated by the method of the gravity center, in which the *n*-th value of the output attribute is calculated by the following expression:

$$Y_n^w = \frac{\int_{min}^{max} \widetilde{X_n^w} * \mu\left(\widetilde{X_n^w}\right) dx}{\int_{min}^{max} \left(\widetilde{X_n^w}\right) dx}, \qquad (11)$$

where $\mu\left(\widetilde{X_n^w}\right)$ is the membership function of a corresponding fuzzy set $\widetilde{X_n^w}$; min and max are borders of the universe of fuzzy variables of *w*-th packet of *n*-th attribute (in our case for the attributes *X7, X8*: *min* – 0, *max* – 65535); $Y_n^w$is the matrix of results of defuzzification for *w*-th packet.

The choice of the method of gravity center is made on the basis of such advantages as separation of the control solution from the statement, application of the universal method, use of the already worked and proven apparatus of fuzzy logic.

This method is the least demanding for computing resources, so their use is useful in the considered application field.

4.5 CONVOLUTION

We have defined the following output parameters of the pre-processing stage (input parameters of the classification stage are the neural network input):

- $Y1_w$ is the value of the protocol state at transmission of the *w*-th packet;
- $Y2_w$ is the belongingness of the source port number to the value related TLS protocol of the *w*-th packet;
- $Y3_w$ is the belongingness of the length of the network packet to the value related TLS protocol of the *w*-th packet;
- $Y4_w$ is the belongingness of the value of integer numbers ContentType of the PayLoad field defined in RFC for protocols TLSv1.0, TLSv1.1, TLSv1.2, and TLSv1.3 of the *w*-th packet.

The number of input parameters is reduced compared to the number of attributes in order to reduce the dimension of the artificial neural network. The procedure for reducing the dimension of an attribute input space is to apply fuzzy arithmetic rules over sets $\mathrm{Y_w}$. We applied:

$$Y2_w = Y_7^w + Y_8^w, \qquad (12)$$

$$Y3_w = Y_4^w + Y_9^w, \qquad (13)$$

$$Y4_w = Y_{10}^w + Y_{11}^w\,. \qquad (14)$$

Besides the specified parameters, the values of attributes X7 and X8 are supplied to the neural network input to improve classification quality.

5. NEURAL NETWORK PROCESSING FOR CLASSIFICATION

The neural network architecture which is most approved now is the multilayer network of direct distribution (called also as the multilayer perceptron) was offered in [34], and gained development in [22, 26]. At the same time, we effectively applied logistic regression to identify the belongingness of network packets to the protected protocols. The method of logical regression allows one to receive probabilistic estimates of the protocol classification.

Having made numerous experiments, the architecture of the multilayer network of direct distribution with one buried layer, which includes the *L* neurons *N*, was selected. Having applied the genetic algorithm [22], we defined quantity of neurons of the buried layer for classification of the protected protocols. So, for TLSv1.1 *L* = 11 neurons, for TLSv1.2 *L* = 12 neurons.

By method of gradient descent we provided training of neural network [35] that allowed one to receive high quality of classification of protocols for the smallest time of calculations.

In solving the classification problem, the sum of the input signals of the hidden layer is converted into the output of the neuron by means of an activating non-linear function σ, which does not possess a memory:

$$N_l = \sigma\begin{pmatrix} \omega_0^l + \omega_1^l Y1 + \omega_2^l Y2 + \omega_3^l Y3 + \\ \omega_4^l Y4 + \ \omega_5^l X_7 + \omega_6^l X_8 + \omega_7^l X_{12} \end{pmatrix}. \quad (15)$$

The choice of activation function σ depends on specifics of a solvable applied task. In this work as activation function a sigmoidal function was applied:

$$\sigma(x) = \frac{1}{1+e^{-x}}\,. \quad (16)$$

The impact of different types of activation functions on the quality of classification is the next stage of research.

6. NETWORK PACKET CLASSIFIER IMPLEMENTATION AND EXPERIMENTAL RESULTS

A comparative analysis of machine learning algorithms is given in [27, 28, 36, 37]. In order to obtain an assessment of the effectiveness of the proposed approach, the software was developed and a bench environment was implemented.

For development of the software, it is possible to use different processing methods and hardware-software solutions now, for example:

- Programming language Python with its rich and rapidly updating arsenal of libraries related to machine learning methods (Scikit-learn [38, 39], Keras [40], Spark [41, 42], TensorFlow [43-45], Theano [46])
- Programming languages C and C++ and specialized libraries with machine learning algorithms such as OpenCV [47, 48]
- Application of hardware solutions on the basis of field-programmable gate array (FPGA) processors (for example, libraries of programming from the Intel company [49, 50]), graphic accelerators of graphics processing unit (GPU) [35, 51], the application-specific integrated circuits (ASIC) (for example, Tensor Processing Unit [44-45]).

However, in our work for the purpose of ensuring cross-platform under different processors and operating systems, the classification program specially developed in C++ was used. Besides, the programming module in programming language C++ was developed to check the presented mathematical apparatus.

The validation of the developed software was realized on the hardware platform with the following characteristic:

- Central processing unit: Intel Core i5-6400 2,7 GHz
- RAM: 8 Gb
- Operating system: MS Windows 10 Pro 64 bit
- Network interface - 100 Mbps.

The architecture of the bench environment implemented is depicted in Fig. 3. During testing on a smartphone and computers with installed applications (Viber, WhatsApp, Google Chrome), information was exchanged with the relevant services via the Internet. The traffic generated by the applications was mirrored to the server, recorded in the form of a dump. Next, a traffic dump was sent to the traffic classification software module.

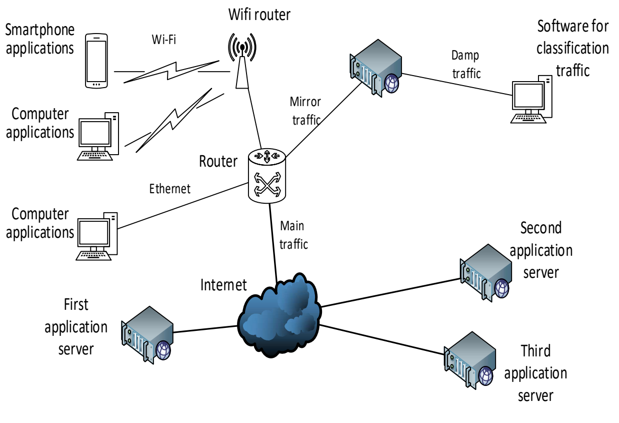

Figure 3 – The architecture of the bench environment

As a training sequence, 1250 packets were sent to the module, and as a testing sequence, we used a series of six dumps with a total length of 9128 packets.

As a training sequence, 1250 packets were used; and as a testing sequence, 1996 packets were sent to the module. The prepared sets of network packets for training and testing had the distribution presented in Fig. 4.

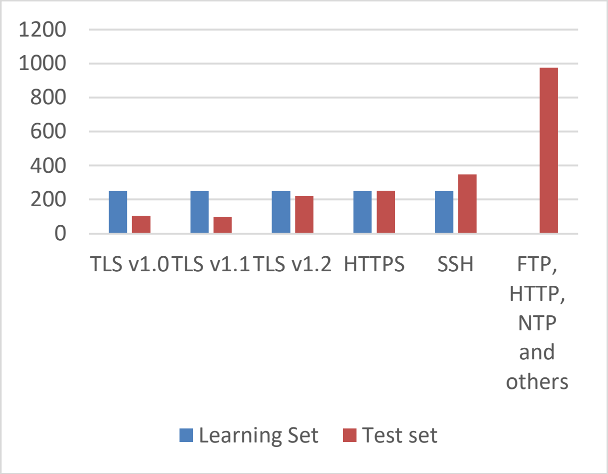

Figure 4 – Distribution of sets of network packets prepared for training and testing

The test results for evaluating the testing time are presented in Table 3.

Table 3. Estimation of classification time for sets of network packets during testing

№	The number of packets in the dump	Classification Time (s)	Average classification time for one network packet (ms)
1	1996	1.21842	0.61043
2	1460	0.85272	0.58405
3	1400	0.84200	0.60143
4	1448	0.89171	0.61582
5	1384	0.84096	0.60763
6	1440	0.87280	0.60611
Total	**9128**	**5.51861**	**0.60425**

Thus, during the testing of the developed classifier, the following results were obtained:

- Probability of a recognition error of the second kind for the protected protocols (TLSv1, TLSv2, etc.) is not less than 0.95%

- Probability of a recognition error of the first kind for the protected protocols (TLSv1, TLSv2, etc.) is not more than 0.05 %

- Average time of classification of the protocol is equal to 0.6 milliseconds.

These results suggest that the developed classifier has, on the one hand, a rather high speed of operation and, on the other hand, a rather high quality of classification. In this case, when creating a classification software module, optimization mechanisms for executable code were not used and hardware accelerators were not involved.

7. CONCLUSION

The technique of classification of the protected application-layer protocols for information exchange, presented in the work, illustrated a modern approach on application of fuzzy logic and neural networks. This approach can be applied to create the efficient information security support systems (IDS, IPS, NMS, DDoS PS, etc.). This approach significantly differs from the algorithms of classification based on the analysis of the sequences of the rules which are previously prepared by highly qualified specialists in information security field.

The main advantages of the approach suggested are a high computational performance of classification and a high quality of classification.

The practical results achieved in testing the suggested technique make it possible to put forward a hypothesis about the possibility of moving away from routine methods of building chains of rules based on signatures, to the construction of adaptive self-configured systems for classification of IP packets of secure application layer protocols, based on methods of fuzzy sets and neural networks. Taking into account the avalanche-like growth of application level protocols, the application of the presented technique in the software of secure information systems will reduce the requirements for the knowledge of system administrators and increase the efficiency of protection of information resources. This direction can be considered as the main direction of further research.

Funding: Research is carried out with support of Ministry of Education and Science of the Russian Federation as part of Agreement No. 05.607.21.0322 (identifier RFMEFI60719X0322).

Conflicts of Interest: The authors declare that there is no conflict of interest.

7. REFERENCES

[1] T.T.T. Nguyen, G.A. Armitage, "Survey of techniques for internet traffic classification using machine learning," *IEEE*

Communications Surveys & Tutorials, vol. 10, issue 4, pp. 56-76, 2008.

[2] H. Kawai, S. Ata, N. Nakamura, I. Oka, “Identification of communication devices from analysis of traffic patterns,” *Proceedings of the 13th IEEE International Conference on Machine Learning and Applications*, Tokyo, Japan, November 26-30, 2017, pp. 1-5.

[3] V. Carela-Español, *Network Traffic Classification: From Theory to Practice*. Universitat Politècnica de Catalunya Barcelona Tech Department d’Arquitectura de Computadors, Barcelona, 2014.

[4] M. Pietrzyk, *Methods and Algorithms for Network Traffic Classification*, PhD Thesis, Telecom Paris Tech Thesis, 2011.

[5] G. Sun, T. Chen, Y. Su, Ch. Li, “Internet traffic classification based on incremental support vector machines,” *Mob. Netw. Appl.*, vol. 23, issue 4, pp. 789-796, 2018.

[6] A.A. Branitskiy, I.V. Kotenko, “Analysis and classification of methods for network attack detection,” *SPIIRAS Proceedings*, vol. 2, issue 45, pp. 207-244, 2016.

[7] R.A. Demidov, A.I. Pechenkin, P.D. Zegzhda, M.O. Kalinin, “Application model of modern artificial neural network methods for the analysis of information systems security,” *Automatic Control and Computer Sciences*, vol. 52, issue 8, pp. 965-970, 2018.

[8] O. Mula-Valls, *A Practical Retraining Mechanism for Network Traffic Classification in Operational Environments*, Master Thesis, Universitat Politecnica de Catalunya, 2011.

[9] A. Saied, R.E. Overill, T. Radzik, “Detection of known and unknown DDoS attacks using artificial neural networks,” *Neurocomputing*, vol. 172, pp. 385-393, 2016.

[10] C. Jie, F. Zhiyi, “Network traffic classification using genetic algorithms based on support vector machine,” *International Journal of Security and Its Applications*, vol. 10, issue 2, pp. 237-246, 2016.

[11] R. Xu, D. Wunsch, “Survey of clustering algorithms,” *IEEE Trans. Neural Networks*, vol. 16, issue 3, pp. 645-678, 2005.

[12] A.I. Getman, Yu.V. Markin, E.F. Evstropov, D.O. Obydenkov, “A survey of problems and solution methods in network traffic classification,” *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 3, pp. 117-150, 2017.

[13] Y.-S. Lim, H.-Ch. Kim, J. Jeong, Ch.-K. Kim, T. T. Kwon, Y. Choi, *Internet Traffic Classification Demystified: On the Sources of the Discriminative Power*, 2010, [Online] Available at: http http://conferences.sigcomm.org/co-next/2010/CoNEXT_papers/09-Lim.pdf.

[14] M. Rehak, M. Pechoucek, M. Grill, J. Stiborek, K. Bartos, P. Celeda, “Adaptive multiagent system for network traffic monitoring,” *IEEE Intelligent Systems*, vol. 24, issue 3, pp. 16-25, 2009.

[15] M.-Y. Liao, M.-Y. Luo, Ch.-S. Yang, C.-H. Chen, P.-C. Wu, Y.-C. Chen, “Design and evaluation of deep packet inspection system: A case study,” *Networks, IET*, vol. 1, pp. 2-9, 2012.

[16] R. Bendrath, M. Mueller, “The end of the net as we know it? Deep packet inspection and internet governance”, *New Media & Society*, vol. 13, issue 7, pp. 1142-1160, 2011.

[17] J. Singh, 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, pp. 43-49, 2013.

[18] S. Abraham, S. Nair, “Cyber security analytics: A stochastic model for security quantification using absorbing Markov chains,” *Journal of Communications*, vol. 9, issue 12, pp. 899-907, 2014.

[19] L.A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, issue 3, pp. 338-353, 1965.

[20] L.A. Zadeh, “Fuzzy algorithms,” *Information and Control*, vol. 12, issue 2, pp. 94-102, 1968.

[21] I. Kotenko, I. Saenko, S. Ageev, “Applying fuzzy computing methods for on-line monitoring of new generation network elements,” *Proceedings of the Third International Scientific Conference “Intelligent Information Technologies for Industry”. Advances in Intelligent Systems and Computing*, vol. 874, Springer, Cham, 2018, pp. 331-340.

[22] A. Piegat, *Fuzzy Modeling and Control*, Springer, 2014.

[23] H. Lim, J. Kim, J. Heo, K. Kim, Y. Hong, Y. Han, “Packet-based network traffic classification using deep learning,” *Proceedings of the 2019 International Conference on Artificial Intelligence in Information and Communication*, 2019, pp. 046-051.

[24] S. Rezaei, X. Liu, “Deep learning for encrypted traffic classification: An overview,” *IEEE Communications Magazine*, vol. 57, issue 5, pp. 76-81, 2019.

[25] T. Terano, K. Asai, M. Sugeno (eds.), *Applied Fuzzy Systems*, Omsya, Tokyo, 1989.

[26] C.M. Bishop, *Neural Networks for Pattern Recognition*, Department of Computer Science and Applied Mathematics Aston University Birmingham, UK, 1995.

[27] K. Dias, M. Pongelupe, W. Caminhas, L. Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, pp. 143-157, 2019.

[28] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapè, "Mobile encrypted traffic classification using deep learning," *Proceedings of the IEEE/ACM Network Traffic Measurement and Analysis Conference*, Vienna, 2018, pp. 1-8.

[29] *Service Name and Transport Protocol Port Number Registry*, 2020 [Online]. Available at: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[30] *The Transport Layer Security (TLS) Protocol. Version 1.2*, 2008 [Online]. Available at: https://tools.ietf.org/html/rfc5246.

[31] E.H. Mamdani, S. Assilian, "An experiment in linguistic synthesis thesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, issue 1, pp. 1-13, 1975.

[32] E.H. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," *International Journal of Man-Machine Studies*, vol. 8, pp. 669-678, 1976.

[33] E.H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, vol. 26, issue 12, pp. 1182-1191, 1977.

[34] M.L. Minsky, S. Papert, *Perceptrons: An Introduction to Computational Geometry*, Cambridge, MA, MIT Press, 1987.

[35] S. Raschka, *Python Machine Learning*, Kindle Edition, 2016.

[36] M. Soysal, E. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Perform. Eval.*, vol. 67, pp. 451-467, 2010.

[37] Y. Okada, S. Ata, N. Nakamura, I. Oka, "Comparisons of machine learning algorithms for application identification of encrypted traffic," *Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops*, Honolulu, HI, 2011, pp. 358–361.

[38] S. Osowski, *Sieci Neuronowe do Przetwarzania Informacji*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, Polsha, 2000. (in Polish)

[39] R. Garreta, G. Moncecchi, *Learning scikit-learn: Machine Learning in Python*. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2013.

[40] A. Gulli, S. Pal, *Deep Learning with Keras*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017.

[41] H. Karau, A. Konwinski, P. Wendell, M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*, O'Reilly Media, Inc., 2015.

[42] N. Pentreath, *Machine Learning with Spark*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2015.

[43] N. McClure, *TensorFlow Machine Learning Cookbook*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017.

[44] R. Bonnin, *Building Machine Learning Projects with TensorFlow*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2016.

[45] S. Abrahams, D. Hafner, E. Erwitt, A. Scarpinelli, *Tensorflow for machine intelligence*, Bleeding Edge Press, Santa Rosa, CA 95404, 2016.

[46] C. Bourez, *Deep learning with Theano*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017.

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

[48] M. Beyeler, *Machine Learning for OpenCV*, Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2017.

[49] *Introduction to Intel® Deep Learning Deployment Toolkit*, 2020 [Online]. Available at: https://docs.openvinotoolkit.org/latest/_docs_IE_DG_Introduction.html.

[50] J. Dean, *Machine Learning for Systems and Systems for Machine Learning*, 2017 [Online]. Available at: https://buzzrobot.com/machine-learning-for-systems-and-systems-for-machine-learning-41438c234e10.

[51] T. Dettmers, *Which GPU(s) to Get for Deep Learning: My Experience and Advice for Using GPUs in Deep Learning*, 2019 [Online]. Available at: https://timdettmers.com/2019/04/03/which-gpu-for-deep-learning/.

Vyacheslav Efimov *obtained the Ph.D degree in 2005. He currently runs the Science and Technology Center at Masshtab Research Institute. His research interests include networks, information security, management systems, systems engineering, information technologies, and systems analysis and synthesis, data mining, artificial intelligence.*

Igor Kotenko *obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Head of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. He has a high experience in the research on computer network security and participated in many projects on developing new security technologies. He was a project leader in the research projects from the US Air Force research department, via its EOARD (European Office of Aerospace Research and Development) branch, EU FP7 and FP6 Projects, HP, Intel, F-Secure, etc. His research results were tested and implemented in more than fifty Russian research and development projects. His main research interests are innovative methods for network intrusion detection, simulation of network attacks, vulnerability assessment, verification and validation of security policy, etc. He has chaired several International conferences and workshops, and serves as editor on multiple editorial boards.*

Igor Saenko *obtained the Ph.D. degree in 1992 and the National degree of Doctor of Engineering Science in 2002. He is Professor of computer science and Leading Researcher of the Laboratory of Computer Security Problems of St. Petersburg Institute for Informatics and Automation. His main research interests are security policy management, access control, management of virtual computer networks, knowledge modeling soft and evolutionary computation, information and telecommunication systems.*