Intrusion Detection System Modeling Based on Learning from Network Traffic Data

Admir Midzic[1], Zikrija Avdagic[2] and Samir Omanovic[3]
[1,2,3] Faculty of Electrical Engineering, Campus of the University of Sarajevo, Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina
[e-mail: admir.midzic@bih.net.ba; zikrija.avdagic@etf.unsa.ba; samir.omanovic@etf.unsa.ba]
*Corresponding author: Admir Midzic

Received December 4, 2017; revised April 20, 2018; accepted July 3, 2018; published November 30, 2018

Abstract

This research uses artificial intelligence methods for computer network intrusion detection system modeling. Primary classification is done using self-organized maps (SOM) in two levels, while the secondary classification of ambiguous data is done using Sugeno type Fuzzy Inference System (FIS). FIS is created by using Adaptive Neuro-Fuzzy Inference System (ANFIS). The main challenge for this system was to successfully detect attacks that are either unknown or that are represented by very small percentage of samples in training dataset. Improved algorithm for SOMs in second layer and for the FIS creation is developed for this purpose. Number of clusters in the second SOM layer is optimized by using our improved algorithm to minimize amount of ambiguous data forwarded to FIS. FIS is created using ANFIS that was built on ambiguous training dataset clustered by another SOM (which size is determined dynamically). Proposed hybrid model is created and tested using NSL KDD dataset. For our research, NSL KDD is especially interesting in terms of class distribution (overlapping). Objectives of this research were: to successfully detect intrusions represented in data with small percentage of the total traffic during early detection stages, to successfully deal with overlapping data (separate ambiguous data), to maximize detection rate (DR) and minimize false alarm rate (FAR). Proposed hybrid model with test data achieved acceptable DR value 0.8883 and FAR value 0.2415. The objectives were successfully achieved as it is presented (compared with the similar researches on NSL KDD dataset). Proposed model can be used not only in further research related to this domain, but also in other research areas.

Keywords: intrusion detection, learning from data, clustering, classification

The model that is presented in paper "Intrusion Detection System Modeling Based on Neural Netwok and Fuzzy Logic" appeared in 2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES), 30 June-2 July 2016, Budapest (Hungary) had shortcomings. They are resolved in the architecture proposed in this manuscript. The main improvement is related to optimization in the second SOM layer using an improved algorithm while keeping acceptable level of DR and FAR. Furthermore, the amount of ambiguous traffic samples that are forwarded to FIS for further analysis is minimized.

http://doi.org/10.3837/tiis.2018.11.022 **ISSN : 1976-7277**

1. Introduction

The amount of information used or exchanged between individuals and companies is increasing very fast. Information is important for individuals and for companies and it has to be protected adequately, whether they are stored locally, in a cloud [1], or transmitted through a different kinds of public or private networks. There are a lots of measures that can be applied to contribute in the process of securing electronic information, such as disabling unauthorized access to information, blocking known attacks, education of users, and all of these can be observed as preventive. To achieve an acceptable level of security in information exchange, detection and prevention of intrusions into computer networks is very important [2]. Intrusion Detection Systems (IDS) try to identify both successful and unsuccessful attempts to abuse computer systems or networks in order to detect intrusions. Source of information which can be, for instance, computer system or a network in which the intrusion is manifested, must be identified [3]. Primary focus of IDS is to identify potential incidents. Some important characteristics of IDS, according to [4] and [5], are the following: *accuracy of prediction*, *performance*, *fault tolerance*, *scalability*, *ability of dynamic reconfiguration* and *configurability*. Intrusion Prevention Systems (IPS) with its architecture and the primary purpose of preventive function cannot exclude need for detection. IPS is not capable providing absolute prevention [6], [7]. There are several classification models for IDS (as shown in **Fig. 1**) [8] and for the research presented in this paper the way in which the analysis is realized is of particular importance. Two approaches for analyzing events in order to detect attacks [9]: m*isuse intrusion detection* and a*nomaly-based intrusion detection* are often combined to obtain better features than used separately [10].

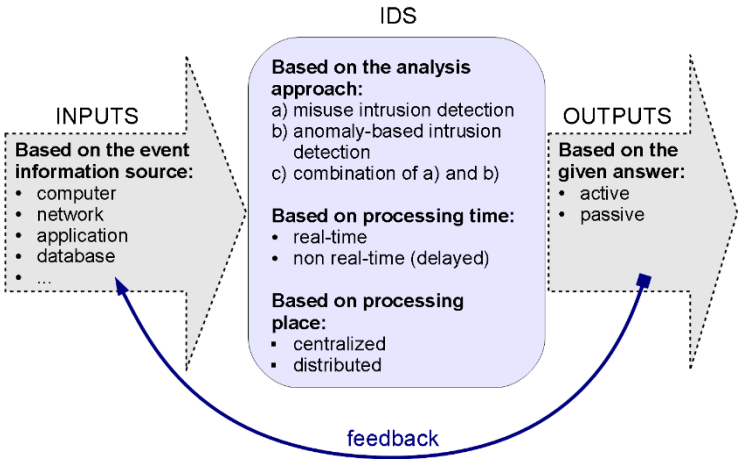

Fig. 1. Classification of Intrusion detection systems

Characteristics of previously described IDS, which are of special importance for the development of the architecture presented in this study, are: *accuracy of prediction*, *performance* and *dynamic reconfiguration*. Accuracy of prediction is important for recognition of those attacks that are represented with a small percentage in the network traffic. Intrusion detection at early stages of the analysis can significantly affect the performance of the system. As network traffic is changing in time, the IDS must be able to learn and

dynamically reconfigure. That is why application of methods and techniques of artificial intelligence is important for creation of IDS with all specified features. This research shows that combination of neural networks and fuzzy logic can be efficient in creation of IDS with previously described features in balance – acceptable level of accuracy and ability of dynamic reconfiguration. Separation and propagation of the ambiguous traffic for further analysis shortens analysis of the majority of network traffic. On the other hand, quantity of separated traffic is relatively small comparing to overall traffic which enables more complex analyses.

2. Related Work

The main motivation behind the research presented in this paper is usage of machine learning from network traffic data in the automation of IDS modeling (optimization of model structure – learning complex rules automatically, optimization of performance, etc.). The main challenge for this model is to sucessufully detect attacks that are either unknown or that are represented with very small percentage of samples in training dataset. This is usually real situation that security operation centers need to have - early notification for suspicious activities (activities that can be intrusions). So, it is important to have agile approach for detecting traffic that is ambigous while keeping acceptable level of detection rate and false alarm rate. Machine learning techniques are especially often used during past years for development of IDS. According to research [11], in the first generation of intrusion detection systems the accent was on single computer systems. That was the time (1970s and early 1980s) when audit records of the operating system were post-processed and both, anomaly detection and misuse detection approaches, were developed [12]. Intrusion Detection Expert System (IDES) project started in 1984. IDES was developed between 1984 and 1986 and it was one of the most important IDS research projects. Results of this project were presented in Dorothy Denning's paper elaborated in 1987 [3] which actually marked beginning of the second phase. Report [13] announced in 1988 pointed out that the statistical profile and expert system approaches to intrusion detection addressed different threats. The IDES prototype was capable of detecting anomalous behavior and reporting anomalies in real time. At the beginning of the 1990, a number of IDSs were developed, mostly relying on a combination of statistical and expert systems approaches [14]. According to [15] the processing is more statistically sophisticated and simple - real-time alerts became possible. The use of competitive neural networks in researches related to intrusion detection in computer networks is present in a number of studies [16]. The same situation is with the application of fuzzy logic, where recent researches have successfully used the advantages offered by this technique [17]. These are the reasons which have determined the architecture used in the study presented in this paper. Self-Organizing Map (SOM) neural networks have been used in the early 80s of the last century [18], and the implementation of the Self-Organizing Feature Map (SOFM) is done through application of classical SOM networks, considerable number of researches have been done with architectures that use SOM in different forms, such as Hierarchical Self-Organizing Map (HSOM) [19], Growing Hierarchical Self-Organizing Map (GHSOM) [20]. GHSOM is the model with the hierarchical structure made of independent, growing SOMs. Some of the researches are primarily focused on exploitation of the advantages offered by these technique and used independently (possibly in cascade and/or parallel connection), but many were focused on the combination of them with other techniques (especially from the field of artificial intelligence) by exploiting multi-layer architectures or forming different types of hybridization [1], [5], [6], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31]. Also, a number of researches are carried out using other techniques (not SOM and not fuzzy logic)

where the dominant role is on those other techniques [32], [25], but very often they use fuzzy logic or SOM for additional fine tuning [15]. Common for all previously mentioned researches is the use of techniques of the artificial intelligence, where the systems based on the application of neural networks with competitive learning [33] and fuzzy logic play an important role. **Table 1.** gives an overview of the most important IDS related publications for this research where fuzzy logic and/or neural networks where used.

Table 1. Overview of the most important IDS related publications for this research

Method(s)	Publications
Self-Organizing Map (SOM)	[16], [19], [20], [33]
Fuzzy logic	[17]
Hybridizations like Adaptive Neuro-Fuzzy Inference System (ANFIS)	[1], [21], [34]

Comparison of IDSs can be done in many different ways. Typically, IDS prediction performance estimation includes Detection Rate (DR) and False Alarm Rate (FAR) (see **Table 2**).

Table 2. DR and FAR measures of performance calculated using confusion matrix

	Predicted: NORMAL	**Predicted:** ATTACK	
Real value: NORMAL	True Negative (TN)	False Positive (FP)	FAR=FP/(FP+TN)
Real value: ATTACK	False Negative (FN)	True Positive (TP)	DR=TP/(TP+FN)

DR is defined as a ratio of number of correctly detected attacks and total number of attacks. FAR is defined as a ratio of number of normal connections wrongly classified as attacks and total number of non-attacks. DR and FAR measures of performance can be calculated using confusion matrix that contains results of: True Negative (TN), False Positive (FP), False Negative (FN) and True Positive (TP). Besides DR and FAR cost matrix is often used for this purpose. Cost matrix enables assigning different misclassification weights to elements of the confusion matrix. For example, misclassification cost matrix for researches based on KDD CUP 99 dataset is presented in **Table 3.** Cost matrix – C and confusion matrix – CM are used to calculate Cost per Example (CPE) value, using (1).

$$\text{CPE}=\frac{1}{N}\sum_{i=1}^{5}\sum_{j=1}^{5}CM(i,j)C(i,j) \quad (1)$$

Matrices C and CM have n^2 members where n represents number of different classes.

Table 3. Misclassification cost matrix

		Predicted				
	Type of attack	*0*	*1*	*2*	*3*	*4*
Real	*0*	1	2	2	2	1
	1	0	2	2	2	0
	2	1	0	2	2	1
	3	2	2	0	2	2
	4	2	2	2	0	2

Diagonal elements of CM are representing correct classifications for each class. Non-diagonal elements of CM are misclassified samples. Matrix C can be used to direct optimization of the classification model. Knowledge discovery competition winner has used cost matrix presented in **Table 3** and has achieved CPE value 0.233097 [35].

Previously listed researches are just a group of the most important researches for this manuscript. There are many other researches related to IDS which are not so related with the approach proposed in this manuscript and thus they are not listed. **Table 4**. shows the most important researches in the available literature which are relevant for the model presented in this paper. Those researches are done by using other methods - not only fuzzy logic and neural networks. Among all researches listed in **Table 4**., there are two papers which preceded the research presented in this manuscript. The first one [36] is related to usage of ANFIS, SOM and Subtractive Clustering. It uses reduced number of attributes and shows good results only with small subsets of data, therefore this is the reason why this approach is abandoned. However, it is noted that usage of SOM and ANFIS is a good way for further research. The second paper [34] presents multilayer architecture with SOM and ANFIS which had some shortcomings which are resolved in the architecture proposed in this manuscript and tested on dataset [37],[38] (that is extracted form dataset [39] used in previous research). The model presented in this manuscript has been developed as two step process. In the first step data were separated in two categories: data that SOMs are able to classify – known data and data that SOMs are not able to classify with unique value of traffic type – those data are for the SOMs ambigous (overlapped) – unknown. Tha main objective in this step was to separete smaller portion of unknown data using this agille classification approach. In the second step smaller portion of data (unknown data) were additionaly analysed using fuzzy system developed with ANFIS. The main improvement is related to SOM in second layer (the amount of ambiguous traffic which forwarded to FIS for further analysis which is optimised as well as number of cluster in SOM second layer).

In [34] we have used static value for numer of nodes (25) and that was the number of the neurons in the second layer for each of 32 cluster in first SOM layer. In this manuscript we have used improved version of algorithm for creation the SOM in second layer. This version ensured that system is generating SOM network in second layer with optimal number of clusters (nodes) - size of SOM second layer is optimal. That was done by repeating experiments (as it is shown in Pseudocode 1) using different values in order to determine the best value (that corresponds to each cluster from the first layer). The best value for the number of nodes (cluster) is the one with the minimaln number of unknown type traffic samples in the second layer.

Another improvement is made during FIS creation using another SOM. Fuzzy system [34] is formed by using ANFIS on training data set when system is trained in several iterations. Training data were first divided into different training matrices. Each of these matrices contained its own training data class. Sub-groups with similar samples are created inside of every class (ten for every class). From every of these sub-groups a portion of data has been taken. They are then used to create new training, validation and test matrix as input data for developing fuzzy systems using ANFIS.

Fuzzy system in this manuscript is created from the training data that SOM (First and Second Layer) was unable to classify. Those training data were clustered using SOM neural network whose size is also determined dynamically as it is explained in 3. Methods and data

Table 4. The most important methods, datasets and results relevant for architecture of our model

Method(s) and Dataset	Significance
ANFIS [1], KDD CUP 99	This model used training and test datasets with reduced number of attributes (30 of 41) and achieved good results. The approach presented in this paper used full attribute list from dataset with ANFIS.
K-means clustering [40], NSL KDD	The best results were generated when the number of clusters matches the number of data types in the data set. This fact, in combination with observation from other mentioned researches (primary [19]) is used as recommendation for dimension of the first layer SOM in the architecture proposed in this paper.
SFFS-RF [41], NSL-KDD	Proposed system constructs a feature subset and classification model with the selected feature using a sequential forward floating search (SFFS) and a random forest (RF) for evaluation of the classification accuracy.
Hierarchical SOM [19], KDD CUP 99	The architecture proposed in this paper also uses SOMs, but organized in two layers with sizes of networks determined according to the number of training samples.
Growing Hierarchical SOM (GHSOM) [20], KDD CUP 99	SOM block used in the architecture proposed in this paper has some characteristics of growing SOM - it grows horizontally until defined value is reached.
Fuzziness based algorithm using neural network with random weights [28], NSL KDD	Influence of this approach to the research presented in this paper is related to building hybrid model and finding groups of data that have significant influence on the classifier performance – that way the special treatment for them is ensured.
Fuzzy Logic [17], NSL KDD	The architecture proposed in this paper also contains fuzzy block designed using ANFIS. It is used to classify ambiguous data.
Supervised SOM, Genetic Algorithm (GA) [30], KDD CUP 99	SOM was used to cluster data. After that, from each of the clusters, the assigned attack type to make decision is retrieved. That decision can be final classification result, or data record needs to be re-evaluated.

Method(s) and Dataset	Significance
Fuzzy cognitive maps and SOM [33], KDD CUP 99	Model uses a set of parallel soft computing based classifiers (SOM and FCM) for detecting abnormal behaviors of network data. That is similar to architecture proposed in this paper. The major difference is that ANFIS (FIS) is used after the second SOM layer and is not used for correction, but for classification.
ANFIS, GA, SC [21], KDD CUP 99	A common feature of the architecture proposed in this paper and the architecture presented in [21] is a layered structure.
Fuzzy Clustering (FC), Feed forward ANN [23], NSL KDD	Fuzzy clustering technique is used to generate different training subsets, for different ANNs that are trained to create different models, while a meta-learner, fuzzy aggregation module, is employed to aggregate these results. In the architecture proposed in this paper SOMs are used to cluster training data, and after that to prepare separated, ambiguous data as a new training dataset for ANFIS.
SOM, GA [24], KDD CUP 99	The original SOM for each attack type was used simply to identify if an attack was detected. In the research presented in this paper SOM block is used for clustering of training dataset and separation of clusters with data belonging to only one class from clusters with data belonging to more than one class.
Random Tree [32], NSL KDD	This approach can be combined in an attempt to get better results when working with the complete dataset.
SOM [31], KDD CUP 99	In the research presented in this paper one SOM block is used for clustering of all traffic (not per traffic type as it is used in [31]).
ICLN and SOM [16], KDD CUP 99	Results for ICLN (Improved competitive learning network) and SOM that were output from the research [16] confirmed that SOM is a good choice (as a competitive neural network).
SOM, ANFIS and Subtractive Clustering [36], KDD CUP 99	The algorithm proposed in paper does not have SOM size optimization and it works with reduced number of attributes in ANFIS part. Our further experiments with this architecture [36] didn't give expected results and this architecture is abandoned.
SOM, ANFIS [34], KDD CUP 99	The main difference between our improved algorithm and [34] is that our improved algorithm proposed in this manuscript uses dynamic SOM size determination (optimization) in the second layer.

3. Methods and data

3.1 Data

To perform testing and to be able to measure the performance of the modeled IDS, it is necessary to have appropriate data sets with description of network traffic events (legitimate and illegitimate). With the sponsorship of DARPA, MIT Lincoln Laboratory has organized the event in 1998 where simulated and generated events in isolated environment are presented.

The generation of data is repeated in 1999, but with included information related to the security of network and computer components involved in the simulation environment. The information contained in this data set is generally divided into two groups of data: normal (legitimate traffic) and traffic with attacks (illegitimate traffic). Attacks are categorized into the following categories: *Denial of Service (DoS) attacks*, *Probe* and *Compromise* (these attacks are presented with data grouped into two subcategories: *Remote to Local - R2L and User to Root - U2R*. The data is divided into training and test data set. The data set for training includes twenty four (24) types of attacks. New fourteen (14) types were added to the test data set. KDD CUP 99 is the most commonly used in researches related to intrusion detection [25]. An analysis of the evaluation based on KDD CUP 99 data set showed many shortcomings that lay behind this dataset and that is why the NSL KDD dataset was created [37] ,[38].

The aim was to solve some of the problems previously identified with KDD CUP 99 dataset. The construction of this dataset reduced the need for random selection of data from the original KDD CUP 99 dataset (i.e. selection of smaller groups of data - subsets) as the number of training and testing data is significantly lower than in the original dataset. The modification involved reducing the original data set but also the introduction of an additional attribute. The training dataset is composed of twenty one of different types of attacks while a test dataset is composed of thirty seven attack types (it contains additional sixteen attack types). The known types of attacks are those that are present in the training dataset, while new attacks are present only in the test dataset, they are not present in the training data set (this approach is taken from KDD CUP 99) [39]. In this study for training, files KDDTrain.TXT and KDDTrain+_20Percent.TXT were used (see **Table 5**).

Table 5. NSL KDD content

Files	Description	Number of records
KDDTrain+.TXT	Complete NSL KDD training data set that includes labels for types of attacks and weights for records.	125973
KDDTrain+_20Percent.TXT	20% subset of complete dataset.	25192
KDDTest-21.TXT	Complete dataset without records labeled with 21.	11850
KDDTest.TXT	Complete NSL KDD test data set that includes labels for types of attacks and weights for records.	22544

The testing is done with the file KDDTest.TXT in which data belonging to the normal traffic, data with attacks for which system is trained, and data with new attacks. In our research we used a concept similar to the one used in the labeling of NSL KDD data set [25], [39], except that additional attribute is not introduced. Data are simply spitted in two groups – those with unambiguous mappings and those with ambiguous mappings. For data with ambiguous mappings SOMs are used for additional clustering inside of each traffic group. This way the uniformity of distribution of samples is achieved.

Although, there are a lot of other datasets (that are created after KDD CUP 99 and NSL KDD), we decided to use NSL KDD. KDD CUP 99 aggregates knowledge of many experts in the field of intrusion detection. This is the most widely used dataset in many previous researches related to the intrusion detection systems [25], [38], [39] and [41]. NSL KDD (KDD CUP 99) and it is specific in terms of class distribution (overlapping).

3.2 Methods

Learning can be done from labeled and unlabeled data in different ways. The majority of researches as it is presented in the Related Work section of this paper [16], [19], [20], [24], [26], [28], [30], [31] and it is based on usage of SOM as one of the key modeling element. One of the main features of the SOM is that the network nodes are distributed in space and they are forming groups of similar input vectors (using unlabeled data), while the output nodes compete to be triggered in response to a particular input vector. However, in some situations, such as the overlapping of classes methods like SMOTE: Synthetic Minority Over-sampling Technique can be used [42]. But, it is also possible to apply supervised training on SOM.

In this case the training dataset contains not only input vectors but also the expected output vector (label vector; class). Thus, label vector helps the SOM to achieve better clustering (with no overlaps) as it will be show in this research. How many neurons (nodes) should have a SOM network can be determined using nonlinear multidimensional data projection and visualization (where lighter colors are used for closer models and darker colors indicate farther models), or by using histogram, or simply by trial-and-error experiments [18]. One of the simplest, but also very efficient way to determine the map size (number of neurons, i.e. number of clusters) [43] is using (2):

$$number\ of\ neurons = \sqrt{n} \tag{2}$$

where n is the number of samples of the training dataset. SOM is quantization method, with limited spatial resolution to represent clusters. If there is an overlap or is insufficiently clear separation of neurons, the number of neurons should be increased in order to obtain greater precision, i.e. to obtain the model with the higher learning capability. The hybrid approach in modeling IDS, proposed in this paper is also based on SOM, but it has several specific properties comparing to similar proposals in the available literature. The overall block diagram of the proposed approach is presented on **Fig. 2**. Input are NSL KDD datasets presented in **Table 5**. Practical implementation of this approach is done in Matlab running it on virtualization platform with no hardware (performance) limitations.

The left side of this figure shows the IDS creation process. The right side shows the created elements of the hybrid model. By using the SOM in first layer we have solved problem where only limited number of training samples can be selected (how to decide in what proportion should the classes be represented) [44]. The input datasets, are preprocessed to transform input data into ranges and forms necessary for training of SOM and ANFIS neural networks. Symbolic values are transformed into numerical values and in the end all values are normalized. All preprocessed data are used in the first layer where SOM network is used for clustering. Generally, SOM clustering & comparing block (the first layer with SOM network & the second layer of SOM networks) provides information about disharmony between SOM clustering results and the expected outputs (class labels). All input attributes from the training dataset are used for training.

The column 42 that contains the expected output (label of traffic class) is used to determine situations in which the given SOM output and the expected output are not matched – the result of SOM clustering and the expected output (label) are compared to detect data belonging to different expected traffic classes that are in the same cluster. The SOM is trained with different parameters in repeated experiments. In the proposed IDS modeling, the SOM network size in first layer is 8x4 (it contains 32 neurons), so the value of the *k* is 32. These dimensions are sufficient to encompass the variety of attacks that can occur in the training data set (the number is not greater than 32). At the same time, a large amount of data can be processed within a reasonable time if these dimensions are used. This layer prepares subsets of data for the next layer. These subsets (clusters) are specific and it is necessary to treat them independently. Each of them is further clustered into smaller subsets using one SOM network in the next layer. Number of clusters for SOM networks in the second layer is determined using formula (3).

$$p \approx \frac{\sqrt{n}}{k} \tag{3}$$

where *n* is number of training samples and *k* is the number of clusters in first layer. Application of hybrid approach (using SOM and ANFIS) on KDD CUP 99 data set, with *p* calculated using formula (3) that is published in [34] was with the static value of *p* and with the different architecture, comparing to the one proposed in this paper. Experiments with the hybrid approach presented in **Fig. 2** are performed on the NSL KDD training dataset, and are based on a different calculation of the number *p*. Here is the number *p* determined by repeating experiments, as is presented in Pseudocode 1.

Total number of necessary clusters is determined using formula (3). Number of clusters in the first SOM layer is fixed and it is equal to 32. Then, experiment in the second layer is repeated using different *p* values to determine the best value of *p* that corresponds to each cluster from the first layer. The best *p* value is when number of unknown type in the second layer is minimal. After finding best *p* values for SOMs in the second layer, complete structure of the second layer is created. Clusters in the second layer are analyzed, clusters with classes 0, 1, 2, 3, and 4 are labeled as final, and unknown type is separated to be forwarded to ANFIS.

Pseudocode 2 presents steps within the block for creation of new subsets for training, validation and testing (see **Fig. 2**). Data separated for ANFIS are clustered using SOM to detect outliers and remove them from ANFIS training. Number of clusters is determined using formula (3). Then, from each cluster, data assigned to the least represented classes are removed and those assingment to the most represented classes are used further. Based on the splitting ratio, one significant part of samples is placed into the training subset, one smaller part of samples is placed in the validation subset and also one smaller part of samples is placed in the testing subset. This ensures proper distribution of samples that is needed for correct process of creation of the hybrid model.

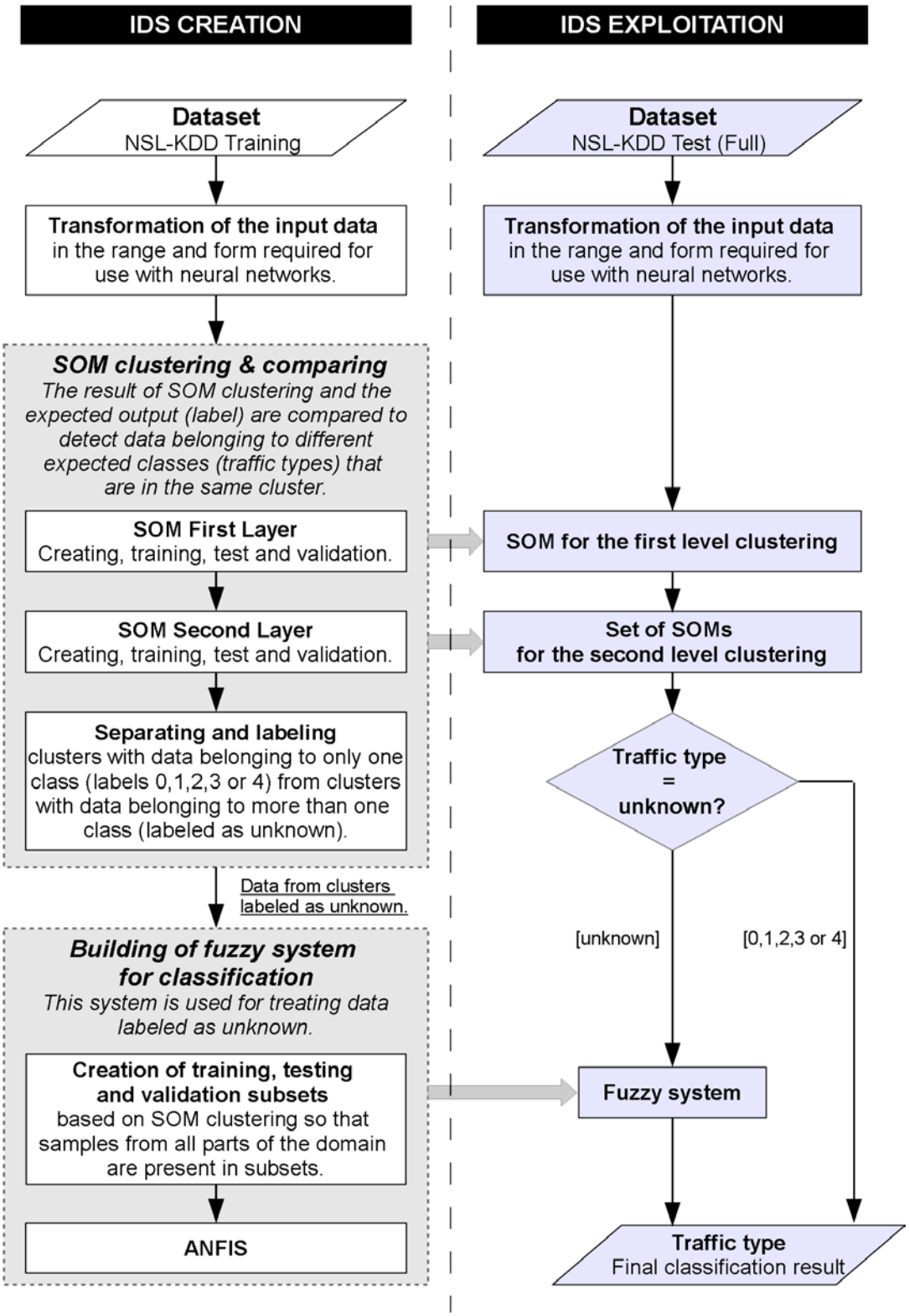

Fig. 2. The hybrid approach in modeling IDS proposed in this paper

Pseudocode 1 – *SOM layers training*

```
trSSet ← Training subset from the entrance of SOM clustering & comparing block on Fig.2.
k ← 32  # Number of neurons for the first layer SOM
# Building and training of SOM network with the k neurons, in the first SOM layer
trSOMFL ← train(selforgmap(k), trSSet) # One SOM for preparing of k Data Clusters
# Calculation of best sizes for SOMs in the second layer
best_p ← zeros(1,32)    # Initial sizes
ii ← 0
FOR EACH datCl IN dataClustersOf(trSOMFL)
        ii ← ii+1
        aa ← round(sqrt(size(datCl))/2)         # Starting value for p
        bb ← round(sqrt(size(datCl)))           # Ending value for p
        min_unknown ← size(datCl)               # Initial count of unknown class
        FOR p FROM aa TO bb STEP 1
                # Training of one SOM in the second layer
                tr1SOMSL ← train(selforgmap(p), datCl)
                [cl0, cl1, cl2, cl3, cl4, unknown] ←
separate(dataClustersOf(tr1SOMSL))
                IF min_unknown > size(unknown) THEN
                        # New best p for the current datCl is found
                        best_p(ii) ← p
                        min_unknown ← size(unknown)
                END IF
        END FOR
END FOR
# Training of SOMs in the second layer using best sizes best_p
FOR EACH datCl IN dataClustersOf(trSOMFL)
        ii ← ii+1
        trSOMSL(ii) ← train(selforgmap(best_p(ii)), datCl)
        [cl0, cl1, cl2, cl3, cl4, unknown] ← separate(dataClustersOf(trSOMSL(ii)))
        labelClustersOf(trSOMSL(ii))
        ADD unknown TO trDataSetForANFIS
END FOR
```

Pseudocode 2 – *Creation of subsets for ANFIS training, validation and testing from* **trDataSetForANFIS**

```
numOfNeurons ← round(sqrt(size(trDataSetForANFIS)))
# SOM clustering of dataset
trSOM ← train(selforgmap(numOfNeurons), trDataSetForANFIS)
FOR EACH datCl IN dataClustersOf(trSOM)
        importantData ← filterTheMostRepresentedClassData(datCl) # Remove outliers
        [tr%, va%, te%] ← assignDataSplittingRatio()
        ADD (tr% samples from importantData) TO trSSet
        ADD (va% samples from importantData, not inluded in trSSet) TO valSSet
        ADD (te% samples from importantData, not inluded in trSSet and valSet) TO
teSSet
END FOR
```

4. Experimental Classification Results and Analysis

The creation of SOM clustering & comparing block and it's main characteristic is shown in **Table 6.** In the second column number of training records placed in each cluster in the first SOM layer is presented.

Table 6. Characteristics of SOM clustering & comparing component (see **Fig. 2**)

Ordinal number of data cluster from the first SOM layer (1,2,...,*k*)	Number of records in the cluster	The best *p* for the cluster	Number of ambiguous records for the cluster, using the best value of *p*
1	621	25	513
2	1822	42	827
3	8843	89	61
4	6163	43	549
5	4052	59	114
6	108	5	0
7	49	4	0
8	5659	47	141
9	1923	42	976
10	2020	38	441
11	52	6	0
12	4355	33	0
13	11129	94	14
14	10	2	0
15	497	21	47
16	2231	24	0
17	2964	54	210
18	1744	30	848
19	4348	57	267
20	2551	48	287
21	1984	43	33
22	1989	23	0
23	1928	22	0
24	17467	66	0
25	1447	38	411
26	7477	73	355
27	1483	39	69
28	3242	50	42
29	3911	32	0
30	5051	36	0
31	3485	30	0
32	15368	62	0
Total number of data cluster in the first SOM is 32	*Total number of training samples (**Table 6.**) is 125973.*	*Total number of data clusters in the second SOM layer* is *1277*	*Total number of ambiguous records is 6205.*

Third column shows the best *p* values – the best number of clusters that should be used for the SOM in the second layer to cluster data from the corresponding first layer cluster. In the fourth

column is number of ambiguous records that needs further processing. Optimization of *p* values was done to minimize number of ambiguous records. Total number of data clusters in the second SOM layer is 1277. Ambiguous records (6205 records) are further used in ANFIS block. Partial classification results at this point are presented in **Table 7** and **Table 8**. Results of classification of test ambiguous records using FIS are presented in **Table 9**. and **Table 10**.

Table 7. Classification results for SOM Block with distribution per attack type

	Predicted value							
Real value	Traffic type	0	1	2	3	4	unknown	
	0	**8742**	25	538	6	187	0	
	1	50	**981**	40	42	107	0	
	2	338	191	**5395**	0	36	0	
	3	0	0	0	**2**	0	0	
	4	1	0	0	6	**184**	0	
	unknown	580	1224	1485	144	2240	**0**	*5673*
	Total per traffic type	9711	2420	7458	200	2754	0	*22544*

Table 8. Classification results for SOM Block as normal and attack (without attack type)

	Predicted NORMAL	**Predicted** ATTACK	
Real value: NORMAL	**8742**	981	**FAR = 0.0561**
Real value: ATTACK	586	**6562**	**DR = 0.9180**

Table 9. Classification results for FIS block with distribution per attack type

	Predicted value						
Real value	Traffic type	0	1	2	3	4	
	0	**448**	960	756	38	1373	
	1	33	**201**	201	54	8	
	2	57	40	**337**	28	32	
	3	3	5	165	**0**	7	
	4	39	18	26	24	**820**	
	Total per traffic type	580	1224	1485	144	2240	*5673*

Table 10. Classification results for FIS block as normal and attack (without attack type)

	Predicted NORMAL	**Predicted** ATTACK	
Real value: NORMAL	**448**	3457	FAR = 3.8583
Real value: ATTACK	410	**1358**	DR = 0.7681

Total test results of created hybrid IDS are shown in **Table 11**. and **Table 12**.

Table 11. Classification results for best proposed hybrid model with distribution per attack type

		Predicted value					
Real value	Traffic type	0	1	2	3	4	
	0	**9190**	985	1294	44	1560	
	1	83	**1182**	241	96	115	
	2	395	231	**5732**	28	68	
	3	3	5	165	**2**	7	
	4	40	18	26	30	**1004**	
	Total per traffic type	9711	2421	7458	200	2754	*22544*

Table 12. Classification results for best proposed hybrid model as normal and attack (without attack type)

	Predicted NORMAL	**Predicted** ATTACK	
Real value: NORMAL	**9190**	4438	FAR = 0.2415
Real value: ATTACK	996	**7920**	DR = 0.8883

5. Discussion

Researches [17], [28], [32], [40] and [41] (see **Table 4**) are based on NSL KDD dataset and thus they are used for results comparison with the results of this research, while other researches are relevant regarding architecture and methods used. Results presented in this manuscript show that detection of attacks that are present in very small quantities have increased with improved algorithm. So, detection of U2R has increased for 1% and detection of R2L has increased for 8.13%. This is related to the change introduced in SOM Clustering & Comparing block (comparing it to previous research [34]). In the other research [32], attributes of network traffic are analyzed (influence of four attribute groups on DR and FAR). Our research used all traffic attributes without grouping them, but samples were separated during IDS creation to prevent influence on outliers. Thus, majority of samples (125973) are used for creation of two layers of SOMs. Smaller part of ambiguous samples (6205), as it is presented in **Table 6**, are used for creation of FIS block. Results presented in [40] are focused on problems in using k-means algorithm when number of clusters is increasing. The best results are obtained using 22 clusters – TN=12907, FP=542, TN=11524, and TP=219, using 25192 samples from NSL KDD. Research [40] emphasizes the importance of proper selection of number of clusters, i.e. optimization of number of clusters. Usage of SOMs and larger number of clusters (1277), which is presented in our paper, showed the way of optimization of number of clusters (p values in the second SOM layer) and preserving acceptable level of DR and FAR. Research [28] used "divide-and-conquer" strategy to categorize samples using "magnitude of fuzziness". Neural network is used for classification and it shows good learning performances. Research [17] was focused on fuzzy sets and usage of FCM. The detected values (predicted classes) are presented to the system administrator for verification. Research

[41] proposed a feature selection technique for IDS to reduce the FP and to overcome performance problems showing that it is possible to shorten the learning time and detection time. Detection rate (84.4%) and false rate (0.4%) for this model [41] is very close to our model results (presented in this manuscript).

Comparison of our research results with results in papers [17], [28], [32], [40] and [41] is presented in **Table 13.**

Table 13. Results comparison

Research and the most important conclusions related to DR and FAR	*Number of tested samples*	*DR*	*FAR*
Random Tree [32], Basic attributes – high DR, content attributes – high FAR, traffic attributes – low DR, host attributes – low FAR.	22544	0.8078	3.22
k-Means [40], As the number of cluster increases above the number of data types the detection rate decreases while false alarm rate gives good results	25192	0.0186	0.0403
Semi-supervised learning [28], Samples that belongs to the mid fuzziness group have a higher risk of misclassification	22544	0.8412	not presented
Fuzzy Controler [17], Interaction between system-user and IDS, with the aim to verify predictions of the system	11850	0.8671	0.5791
SFFS-RF, Feature selection algorithm proposed for SFFS-RF shows good results in terms of a lower computation cost and higher classification results then the other feature selection techniques. [41]	22544	0.844	0.4
Proposed model - SOM block. Ambiguous samples are forwarded to FIS block, because SOM block is unable to classify them.	16871	0.9180	0.0561
Proposed model - FIS block. Ambiguous samples achieve better DR, but have high FAR.	5673	0.7681	3.8583
Proposed hybrid model (SOM+FIS). Hybrid Model gave comparable results (DR and FAR)	**22544**	**0.8883**	**0.2415**

6. Conclusion

Obtained results, which are comparable with other simillar research (they are not the best for this dataset), show that the proposed hybrid model, successfully classifies samples, despite the fact that classes are not equally represented in the datasets. Final structure of the proposed model has 1277 SOM clusters organized in two layers and they classify majority of traffic, while the small part of ambiguous traffic is forwarded to FIS. SOM clusters that are created during training are analyzed. If samples (all of them) within one cluster are marked as members of the same class, the analyzed cluster is final cluster. Otherwise, samples from cluster are forwarded for further analysis. The information about the expected class is used during SOM training, but in the specific manner. So, the main novelty in the proposed

architecture is improved algorithm for the optimization of second SOM layer while the number of ambiguous samples that are forwarded from second SOM layer to FIS is minimized. IDS is then able to detect even those attacks that are represented with small quantity in overall traffic.

The CPE value of the proposed hybrid model, according to formula (1), misclassification cost matrix in **Table 3** for the best result (presented in **Table 11**) is 0.4281. But, DR and FAR od proposed architecture for majority of the traffic (0.9180 as presented in **Table 8**) are comparable and acceptable with other similiar researches (as presented in **Table 13**) and even for amiguous traffic (as presented in **Table 12**). This fact (high DR and low FAR) and possibility to separate ambigous traffic data samples using our algorithm, are the main contributions of this research.

In the future, we will try to develop an algorithm that will be able to improve ambiguous data samples classification.

References

[1] P. Nagarajan, G. Perumal, "A Neuro Fuzzy Based Intrusion Detection System for a Cloud Data Center Using Adaptive Learning," *The Journal of Institute of Information and Communication Technologies of Bulgarian Academy of Sciences*, vol. 15, no. 3, pp. 88–103, 2015. Article (CrossRef Link)

[2] B. Mukherjee, L. T. Heberlein, Karl N. Levitt "Network intrusion detection," *IEEE Network*, May/June: pp. 26-41, 1994.

[3] J. McHugh, "Intrusion and intrusion detection," *International Journal of Information Security*, vol. 1, no. 1, pp. 14–35, 2001. Article (CrossRef Link)

[4] E. H. Spafford, D. Zamboni, "Intrusion detection using autonomous agents," *Computer Networks, Elsevier*, vol. 34, no. 4, pp. 547-570, 2000. Article (CrossRef Link)

[5] S. Chebrolu, A. Abraham, J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Journal Computers and Security, Elsevier*, vol. 24, no. 4, pp. 295–307. 2005. Article (CrossRef Link)

[6] C. Modi, D. Patel, H. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications, Elsevier*, vol. 36, no. 1, pp. 42-57, 2013. Article (CrossRef Link)

[7] K. A. Scarfone, P. M. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," *Recommendations of the National Institute of Standards and Technology*, 2007. Article (CrossRef Link)

[8] H. Debar, M. Dacier, A. Wespi "Towards a taxonomy of intrusion detection systems," *Computer Networks, Elsevier*, vol. 31, no. 8., pp. 805–822, 1999. Article (CrossRef Link)

[9] A. Lazarevic, V. Kumar, J. Srivastava , "Intrusion Detection: A Survey," *Managing Cyber Threats-Issues, Approaches, and Challenges, Springer: pp. 19-80*, 2005. Article (CrossRef Link)

[10] W. Lee, S. Stolfo, K. Mok, "Adaptive Intrusion Detection: A Data Mining Approach," *Artificial Intelligence Review* , vol. 14, no. 6, pp. 533–567, 2000. Article (CrossRef Link)

[11] AK Jones, RS Sielken, "Computer system intrusion detection: A survey," *University of Virginia. Technical Report*, p. 25, 2000.

[12] J.P. Anderson, "Computer security threat monitoring and surveillance," *James P. Anderson Co. Fort Washington, PA*, 1980.

[13] D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering, IEEE*, vol. 13, no. 2, 1986. Article (CrossRef Link)

[14] T. F. Lunt, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, et. al. "IDES: The enhanced prototype a real-time intrusion-detection expert system," *Computer Science Laboratory SRI INTERNATIONAL*, p. 88, 1988. Article (CrossRef Link)

[15] G. Pang, K M. Ting, D. Albrecht, H. Jin., " ZERO++: Harnessing the Power of Zero Appearances to Detect Anomalies in Large-Scale Data Sets," *Journal of Artificial Intelligence Research*, vol. 57, pp. 593-620, 2016. Article (CrossRef Link)
[16] J. Z. Lei, A. Ghorbani, "Network intrusion detection using an improved competitive learning neural network," in *Proc. of IEEE Proceedings Second Annual Conference on Communication Networks and Services Research, IEEE,* pp. 190-197, 2004. Article (CrossRef Link)
[17] F. Geramiraz, A.S. Memaripour, M. Abbaspour, "Adaptive anomaly-based intrusion detection system using fuzzy controller," *International Journal of Network Security*, vol. 14, no. 6, pp.352-361, 2012. Article (CrossRef Link)
[18] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks, Elsevier*, vol. 37, pp. 52–65, 2013. Article (CrossRef Link)
[19] H. G. Kayacik, A. Zincir-Heywood, M. I. Heywood "A hierarchical SOM based intrusion detection system," *Engineering Applications of Artificial Intelligence, Elsevier,* vol. 20, no. 4, pp. 439-451, 2007. Article (CrossRef Link)
[20] Y. Yang, D. Jiang, M. Xia, "Using improved GHSOM for intrusion detection," *Journal of Information Assurance and Security*, vol. 5, pp. 232- 239, 2010. Article (CrossRef Link)
[21] A. N. Toosi, M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer Communications, Elsevier*, vol. 30, no. 10, pp. 2201–2212, 2007. Article (CrossRef Link)
[22] B. Kavitha, S. Karthikeyan, P. S. Maybell "An ensemble design of intrusion detection system for handling uncertainty using Neutrosophic Logic Classifier," *Knowledge-Based Systems, Elsevier,* vol. 28, pp. 88–96, 2011. Article (CrossRef Link)
[23] G. Wang, J. Hao, J. Ma, L. Huang "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Systems with Applications, Elsevier*, vol. 37, no. 9, pp. 6225-6232, 2010. Article (CrossRef Link)
[24] L. DeLooze, J. Kalita, "Applying soft computing techniques to intrusion detection," *Cyber Security and Information Infrastructure Research Workshop,* pp. 70-99, 2006.
[25] L. Dhanabal, S. P. Shantharajah "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446-552, 2015. Article (CrossRef Link)
[26] M. Jazzar, A. Jantan "A novel soft computing inference engine model for intrusion detection", *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 4, pp. 1-9, 2008. Article (CrossRef Link)
[27] M. Pandaa, A. Abraham, M. R. Patra "A hybrid intelligent approach for network intrusion detection," *Procedia Engineering, Elsevier*, vol. 30, pp. 1–9, 2012. Article (CrossRef Link)
[28] R. A. R. Ashfaq, X. Wang , J. Z. Huang, H. Abbas , Y. L. He "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences, Elsevier*, vol. 378, pp. 484–497, 2017. Article (CrossRef Link)
[29] S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, F. Herrera "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems," *Expert Systems with Applications, Elsevier,* vol. 42, no.1, pp. 193–202, 2015. Article (CrossRef Link)
[30] Z. Jian-Hua, LI Wei-Hua, "Intrusion detection based on improved SOM with optimized GA," *JOURNAL OF COMPUTERS*, vol. 8, no. 6, pp. 1456-1463, 2013. Article (CrossRef Link)
[31] V. Venkatachalam, S.Selvan, "Intrusion detection using an improved competitive learning lamstar neural network," *International Journal of Computer Science and Network Security*, vol. 7, no. 2, pp. 255-26, 2007. Article (CrossRef Link)
[32] P. Aggarwal, S. K. Sharma, "Analysis of KDD dataset attributes - class wise for intrusion detection," in *Proc. of 3rd International Conference on Recent Trends in Computing 2015 Procedia Computer Science, Elsevier*, vol. 57, pp. 842–851, 2015. Article (CrossRef Link)

[33] P. Lichodzijewski, A. Nur Zincir-Heywood, M. I. Heywood, "Host-based intrusion detection using Self-Organizing Maps," in *Proc. of IJCNN '02. Proceedings of the International Joint Conference on Neural Networks, IEEE,* vol. 2, pp. 1714-1719, 2002. Article (CrossRef Link)
[34] A. Midzic, Z. Avdagic and S. Omanovic, "Intrusion detection system modeling based on neural networks and fuzzy logic," in *Proc. of 2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES), IEEE,* pp. 189-194, 2016. Article (CrossRef Link)
[35] I. Levin, "KDD-99 Classifier Learning Contest LLSoft's Results Overview," *SIGKDD Explorations*, vol. 1, no. 2, pp. 67-75, 2000. Article (CrossRef Link)
[36] Z. Avdagic, A. Midzic, "The effects of combined application of SOM, ANFIS and Subtractive Clustering in detecting intrusions in computer networks," *MIPRO 2014, IEEE,*. pp. 1582-1587., 2014. Article (CrossRef Link)
[37] NSL KDD Dataset [Internet]: Article (CrossRef Link)
[38] J. McHugh, "Recent Advances in Intrusion Detection. RAID 2000. Lecture Notes in Computer Science," *Springer, Berlin*, pp. 145–161, 2000. Article (CrossRef Link)
[39] S. Revathi, A. Malathi "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)* 2(12):pp. 1848-1853. 2013. Article (CrossRef Link)
[40] S. Duque, M. N. Omar. "Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS)," *Procedia Computer Science, Elsevier,* vol. 61, pp. 46–51, 2015. Article (CrossRef Link)
[41] J. Lee, D. Park and C. Lee, "Feature Selection Algorithm for Intrusions Detection System using Sequential Forward Search and Random Forest Classifier," *KSII Transactions on Internet and Information systems*, vol. 11, no. 10, pp.5132-5148, 2017. Article (CrossRef Link)
[42] N. V. Chawla, K. W. Bowyer, L. O., W. P. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357 2002. Article (CrossRef Link)
[43] J. Vesanto, E. Alhoniemi, "Clustering of the Self Organizing Map," *IEEE Transactions on Neural Networks, IEEE*, vol. 11, no. 3, pp. 556 – 500, 2000 Article (CrossRef Link)
[44] G. M. Weiss, F. Provost "Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315-354, 2003. Article (CrossRef Link)

Admir Midzic received BSc. and MSc. degrees at Faculty of Electrical Engineering Sarajevo, University of Sarajevo, Bosnia and Herzegovina. He is conducting doctoral research related to intrusion detection with neural network and fuzzy logic. He has been working in Joint Stock BH Telecom Sarajevo for 15 years at different positions related to information technologies and information security. His latest research interests are related to Cloud Computing and Internet of Things security.

Zikrija Avdagic is a professor at the Faculty of Electrical Engineering Sarajevo, and lectures at Bachelor, Master and Doctoral studies, covering field of Artificial Intelligence and Bioinformatics. He received his BSc., MSc. and DSc. degrees at the University of Sarajevo. He conducted his doctoral research related to knowledge-based real-time systems, at the University of Stuttgart. He has worked for 20 years as a researcher at leading positions in the industry. At the Faculty of Electrical Engineering, Professor Avdagic founded the research group for the Artificial Intelligence, Bioinformatics & Biomedical Engineering in 2007, and the Intelligent Control Laboratory in 2001. In 2007, D.Sc. Avdagic founded the IEEE - Chapter: Computational Intelligence under the IEEE-Bosnia and Herzegovina Section. In addition to many grants (WUS, DAAD, SOROS and UNESCO) in 2001 he was awarded the Fulbright. In 2009 he received University of Sarajevo Award for the Best Professor. His latest research focus is on fusion of artificial intelligence methods into microarray gene bioinformatics and microscopy image processing (AIB laboratory aib.etf.unsa.ba).

Samir Omanovic is associate professor at the Faculty of Electrical Engineering and lectures at all cycles, covering topics related to software engineering, pattern recognition and image processing. He received his BSc., MSc. and DSc. degrees at the University of Sarajevo. He conducted doctoral research entitled Modeling of Fuzzy Neural Systems Based on Coevolutionary Algorithm, at the same University of Sarajevo. He has worked for 10 years in the software engineering industry (on different positions – software developer, team leader, senior software architect, and department leader). The most of his faculty research is in the field of pattern recognition, while the most of his industry expertise is in the field of software engineering. Dr. Omanovic's current research interests are in areas of pattern recognition, computer vision, image processing and advanced software engineering.