 doi:10.20944/preprints202007.0745.v1

Early detection of the advanced persistent threats attacks using performance analysis of deep learning

Javad Hassannataj Joloudari[1,2], Mojtaba Haderbadi[1], Amir Mashmool[1], Mohammad GhasemiGol[1], Shahaboddin Shamshirband[3], Amir Mosavi[4]

[1] Faculty of Engineering, Department of Computer Engineering, University of Birjand, Birjand, Iran
[2] Department of Information Technology, Mazandaran University of Science and Technology, Babol, Iran
[3] Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam
[4] Institute of Automation, Obuda University, 1034 Budapest, Hungary

Abstract

One of the most common and important destructive attacks on the victim system is Advanced Persistent Threats (APT)-attack. The APT attacker can achieve his hostile goals by obtaining information and gaining financial benefits regarding the infrastructure of a network. One of the solutions to detect a secret APT attack is using network traffic. Due to the nature of the APT attack in terms of being on the network for a long time and the fact that the network may crash because of high traffic, it is difficult to detect this type of attack. Hence, in this study, machine learning methods such as C5.0 decision tree, Bayesian network and deep neural network are used for timely detection and classification of APT-attacks on the NSL-KDD data set. Moreover, 10-fold cross validation method is used to experiment these models. As a result, the accuracy (ACC) of the C5.0 decision tree, Bayesian network and 6-layer deep learning models is obtained as 95.64%, 88.37% and 98.85%, respectively, and also, in terms of the important criterion of the false positive rate (FPR), the FPR value for the C5.0 decision tree, Bayesian network and 6-layer deep learning models is obtained as 2.56, 10.47 and 1.13, respectively. Other criterions such as sensitivity, specificity, accuracy, false negative rate and F-measure are also investigated for the models, and the experimental results show that the deep learning model with automatic multi-layered extraction of features has the best performance for timely detection of an APT-attack comparing to other classification models.

Keywords: APT-attack, detection and classification, feature extraction, machine learning, C5.0 decision tree, Bayesian network, deep learning

1. Introduction

Providing information security is one of the main problems of the companies and organizations, and they constantly try to ensure that their data and information are not compromised due to the accidents and attacks [1]. The attacks and activities of the attackers have become more complicated and targeted owing to the progress and growth of the cyberspace. According to Gartner, budgets have risen from $114 billion in 2018 to more than $124 billion in 2019. Information technology security leaders in companies agree to a 72% budget increase in 2020 to take steps such as continuous staff training, awareness and skill enhancement and reduce the damage caused by intrusion into their systems. Today, most of the attacks that threaten companies are targeted and long time, some of which are known as Advanced Persistent Threats (APT) [2]. The term APT was first introduced in 2006 by US Army Air Force specialists regarding unknown intrusion activities [3]. APT attacks are carried out by a group of well-funded attackers with a pre-determined plan to gain access to the confidential information or data of the companies. This attack is a multi-step and persistent attack through which the attacker can remain in the victim system for several months with full awareness [4], [5], [6], [7].
An APT attack has three characteristics [4], [3], which include 1) threats; the ability of the attacker to access confidential information, 2) advanced; using advanced techniques to complete the attack cycle by the attacker, and 3) persistent; the slow process of the attacker to reach the defined goal. Consequently, an APT attack can be favorable

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

for the attacker from three points of view. The first is that the attacker has unlimited time to attack. Second, the attacker can seize unlimited resources, and third, the organizations need to focus on their business strategies rather than spending all of their resources on defensive strategies [5].
Examples of APT attacks that have occurred in recent years are listed below.
EPIC TURLA, which was identified by Kaspersky, aimed to infect the systems of government agencies, state departments, military agencies and embassies in more than 40 countries worldwide [6].
Deep panda was an attack carried out to obtain the information of the staff of the US Intelligence Service, and was probably of Chinese origin. The attackers used the Deep panda code to endanger the information of more than 4 million employees [7].
In addition, a group from Russia known as Fancy Bear, Pawn Storm and Sednit was identified by Trend Micro in 2014 that launched attacks on military and government targets in Ukraine, Georgia, NATO and US defense allies [5].
In an APT attack, attackers use various methods to intrude, two of which are mentioned as follows: I) Zero Day; Attackers identify the weaknesses of a company or an organization and use them to damage the systems [4], II) Targeted phishing; In this method, attackers use infected emails that contain malware to intrude) the systems [4]. In APT attack, attackers try to find the codes of the target systems and programs at the beginning of the task to intrude the systems. As attacks become more complicated, traditional security systems such as firewalls, web and email protectors and scanners are no longer suitable for defending and preventing damages. One of the serious issues and challenges associated with APT attack is lack of a high-precision and real-time detection system. The other challenges are that the attacker is able to invisibly analyze the victim system using structured models and can be placed in the target system for a long time [8], [2].
Therefore, the methods used by researchers to detect APT attacks are as follows.

- Detection models based on machine learning algorithms, including linear support vector machine, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM [8] as a subset of SVM methods as well as Complex tree, Medium tree and Simple tree for decision tree [9].
- Detection models based on mathematical models, such as hidden Markov model [10].
- Methods and approaches for automatic extraction of features using attack graph [11].
- Techniques to reduce false detection, such as Duqu tool [12].
- Detection of all attack steps using tools, such as SpuNge [13].

Although aforementioned schemes can be relatively appropriate detection methods for dealing with APT attacks, they cannot perform timely detection when attacks occur in real-time. In addition, these methods have high false negative and positive rates, which are important criteria that can indicate the effectiveness of a method in correct detection of the attack. None of these methods provide a system model capable of detecting a new attack pattern, high generalizability and high flexibility. Finally, lack of proper process on the data set of the attacks in these methods is quite obvious. Consequently, we decided to examine and investigate machine learning methods such as C5.0 decision tree, Bayesian network and deep neural network on the NSL-KDD data set. As a result, it can be stated that deep neural network method greatly reduces the weaknesses of the mentioned methods and can be a powerful approach to detect an APT attack on the considered data set, since deep learning model provides high detection accuracy (ACC) as well as automatic extraction of the main features of the attack. In other words, according to our latest and greatest knowledge about APT attack detection, we can reasonably argue that among the available methods, deep learning method [14] as an intelligent method that is used today for large data sets in different organizations, provides the best performance.
It is noteworthy that it is the first time that C5.0 decision tree, Bayesian Network and deep learning models are utilized to detect APT attacks on the NSL-KDD data set. In this paper, deep learning model as a proposed model along with Bayesian Network and C5.0 decision tree models is implemented on the relatively large NSL-KDD data set. In brief, the contributes of the article are as follows:

- Improving the detection accuracy by analyzing the data through deep learning model in comparison with C5.0 decision tree and Bayesian classification models.
- Improving deep neural network training by testing the existing data through Maxout method and cross-validation in order to avoid over-fitting and increase generalizability.
- Proposing a 6-layer deep learning model by automatic extracting and selecting the features in the hidden layers of the neural network.

The remaining sections of this paper are organized as follows. We explain related work in Section 2. Section 3 describes our proposed methodology regarding APT attack detection using classification models. The evaluation of the models' performance is accomplished and analyzed in section 4. Section 5 presents the experimental results.

 doi:10.20944/preprints202007.0745.v1

Section 6 represents "Results and Discussion". Finally, we conclude our paper with some suggestion for future research works in Section 7.

2. Related Work

The detection methods of APT attacks that have been introduced up to now, have disadvantages, such as high rate of false detection of the attacks and lack of real-time detection. The APT attack detection methods with different criteria that have been studied by researchers so far have been investigated in the following. Among these methods that have led to better detection of the attacks are machine learning-based methods.

In [8], Ghafir et al., by receiving the network traffic and after analyzing the data, have implemented algorithms such as decision tree, various SVM models, Nearest neighborhood and Ensemble on the data. They have observed that the SVM linear algorithm has the best result with 84.8% accuracy. Finally, they have introduced a system called MLAPT. It is necessary to mention that they have calculated only the accuracy parameter for the algorithms.

In [9], Chu et al. have used the NSL-KDD database to detect the attack and have utilized the PCA method to decrease the size of the classified data set and have concluded that the SVM algorithm with the radial basis function as the kernel has better performance comparing to the classification algorithms such as multilayer perceptron (MLP), decision tree of J48 and Naive Bayes reaching a detection accuracy of 97.22%.

In an APT attack, since the attacker is following the program with great planning and precision, he makes every effort to behave normally on the network so that the detection tools do not notice his presence, and it makes it difficult to detect the attack. However, in [15], Marchetti et al. have proposed a method to detect the infected hosts. The method receives the network traffic and displays a list of infected hosts at the end of the process.

In [14], Bodström et al. have introduced a model based on a theoretical approach or idea regarding APT attacks, and have stated that the APT attack is a persistent and multi-step attack that uses the entire network stream as input. As a result, experiments demonstrate that the deep learning stack that utilizes sequential neural networks achieves a better and more flexible architecture for the APT attack detection.

In [16], Bhatt et al. have proposed a method to predict and detect APT attacks. Due to the fact that this attack is dynamic and can be developed in several directions in parallel, a combination of attack and defense patterns have been used in this model. To implement the procedure, Apache Hadoop has been performed with a logical layer that includes Information Gathering, Weaponization, Delivery, Exploitation, Installation, Command and Control (C2) and Actions steps, and is capable of predicting and detecting APT attacks. Each of these steps is necessary to pursue the goals.

Despite the growth and spread of APT attacks, no specific research and study has been conducted about this attack, and most of the investigations about APT consist the attack patterns and its general information, and automatic detection methods have not been taken into consideration [17], [18], [19], [20]. One of the most vulnerable platforms is mobile phones, which are very popular for attackers [21]. Due to the fact that this attack uses secret and intelligent techniques, and can stay in the system for months, therefore, traditional intrusion detection systems cannot detect these attacks, because they are usually based on pattern or signature and use applications to detect APT [22], [23], [24].

Statistical methods have also been used to detect APT attacks. Hidden Markov model is one of these methods. In [10], Ghafir et al. have developed a system that can be effective in both predicting and detecting APT attacks. The system consists of two parts or sections, the first of which examines the correlation of the warnings, and the second part uses the Markov model to decrypt the attack, and the count of warnings or stages of the APT attack is considered to be 4, and the system can estimate the sequence of attack stages with an accuracy of 91.80%.

In some cases, the introduced methods and tools are not complete, meaning that they may only be able to detect the vulnerability of the environment or network and not be able to detect the attack in the environment, as in [25] that John et al. have proposed a method to investigate whether the network environment is vulnerable to an APT attack or not. This tool allows the network security administrators to check the vulnerability of the network environment after initial configuration and then make changes if necessary.

In [26], Bodström et al. have utilized Observe - Orient - Decide - Act (OODA) loop and Black Swan Theory for detection and identification of APT attacks. In this paper, without manipulating and reducing the features, the network data stream is transferred to the detection process. It has been suggested that in order to better detect an attack, the most important factor in the attack must be identified, which in the case of APT, is communication factor, and in result, the network stream must be recorded to identify the attack.

In [27], Friedberg et al. state that in order to detect an APT attack, the attack detection system requires a large amount of information and input data. In addition, at the end of the detection process, what the detection system presents as a result is highly complex, and it is very difficult for security analysts to understand it).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

3. Proposed Methodology

In this study, we have used RapidMiner simulator for the APT attack detection and classification process. The methodological process is illustrated in Fig. 1.

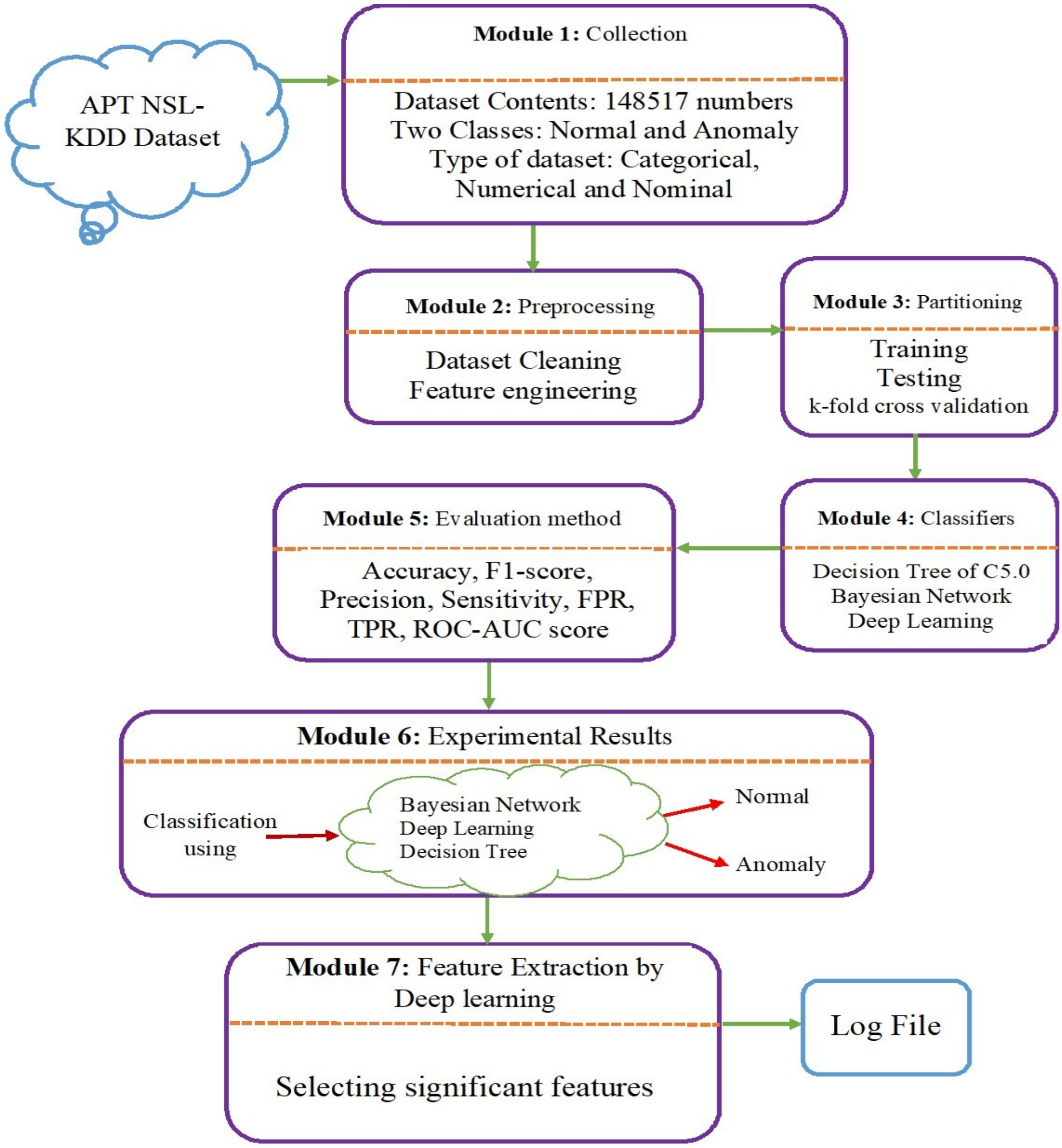

Fig. 1. Proposed methodology.

According to Fig. 1, the proposed methodology includes 7 modules, each of which will be described in detail in the following. In this study, the modules include data collection from an external source, pre-processing, segmentation, classifiers, model evaluation criteria, selection of the best model and extraction of the features.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

3.1. Used Data Set

We used the NSL-KDD data set [9], [28], to detect APT attacks. This data set includes 148517 data samples and includes 125973 training data and 22544 test data. This data set has two normal and anomalous classes; the normal class indicates the normal network status, while the anomaly class indicates the APT attack status.
The data set is shown in Fig. 2 for the two normal and anomalous classes. The data types consist of categorical, numerical and nominal data. Moreover, this data set has 42 features.

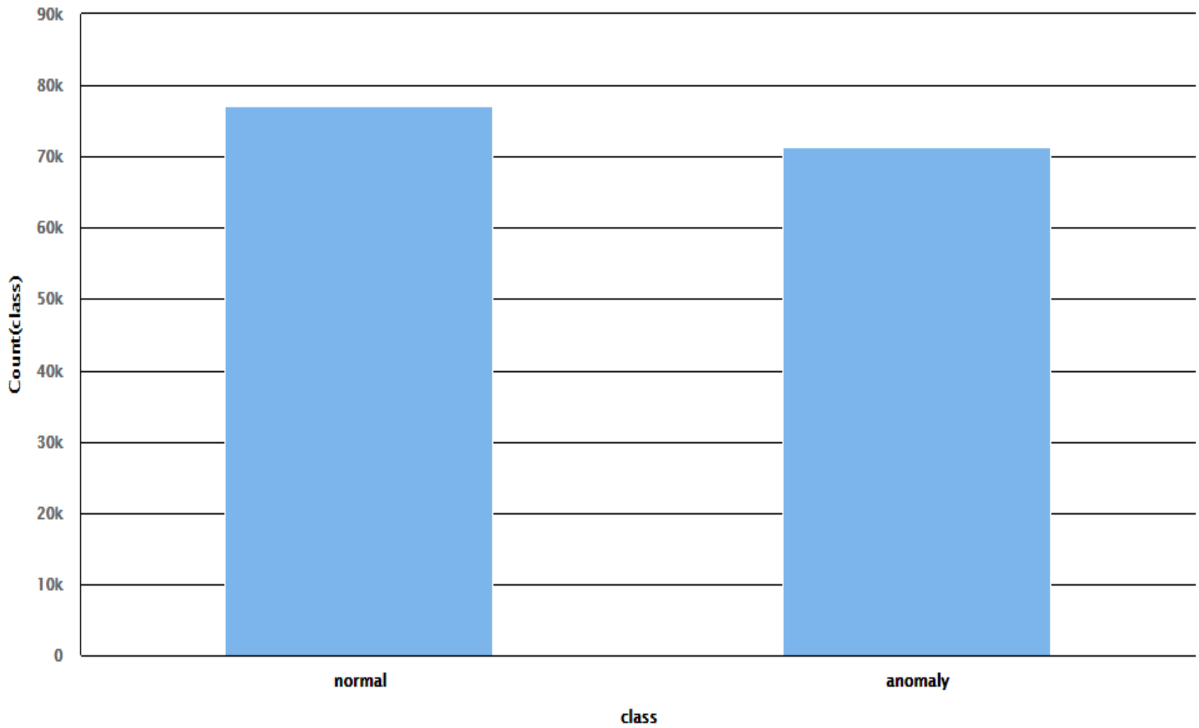

Fig. 2. NSL-KDD data set for two normal and anomalous classes.

3.2. Pre-processing

In this paper, machine learning and deep learning approaches are used. One of the steps in these approaches is pre-processing of the data, which necessitates analyzing the data. In the pre-processing module on the NSL-KDD data set, the data needs to be usable and performable for the classifiers in the next modules. Therefore, for this module, we have investigated the missing data after extracting the samples using RapidMiner software and we have found out that there is no missed and unvalued data. Additionally, we have not used feature selection in this module, since feature selection methods cannot have great effects on analyzing the NSL-KDD data set here. It should be noted that we will investigate and execute feature extraction as an important module in the seventh module. However, feature engineering [29] is utilized in this subsection, i.e. we first need to specify the features of the APT attack in the relevant data set in terms of their type. According to the features of the NSL-KDD data set [30], the data types are as categorical, numerical and nominal data.
It's worth noting that since all the values in the Duration column are zero, we have not put it in any category for the data type and it has been deleted. The NSL-KDD data set also includes two types of classes called Normal and Anomaly. Furthermore, in order to complete the pre-processing of the data, we have aggregated and integrated the training and testing data sets so that the count of normal and anomaly samples is 77054 and 71463, respectively.

3.3. Partition of the NSL-KDD data set

As mentioned in subsection 3.1., the NSL-KDD data set used in this research includes 148517 samples, and as data segmentation, we consider 90% of them for training and 10% for testing. In addition, we have used k-fold cross

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

validation method, and for examining the proposed models, 0.9 of the data is used for training and the remaining 0.1 of the data is used for testing for each fold. In section 5, giving the experiment results, the data classification will be explained in more detail.

3.4. Classification models

3.4.1. C5.0 Decision tree

In the process of improving decision tree models, C5.0 decision tree is the latest generation of the decision tree models, including CHAID, ID3 and C4.5 [31], [32]. The main task of the decision tree is to create rules that can help the security experts to detect the type of the input data based on the constructed model. A decision tree model consists of a number of nodes and branches so that the leaves (external nodes) represent normal and anomalous classes or a set of answers, and in other nodes (internal nodes), the decisions are made based on one or more features. A decision tree diagram with a depth of 5 is shown in Fig. 3.

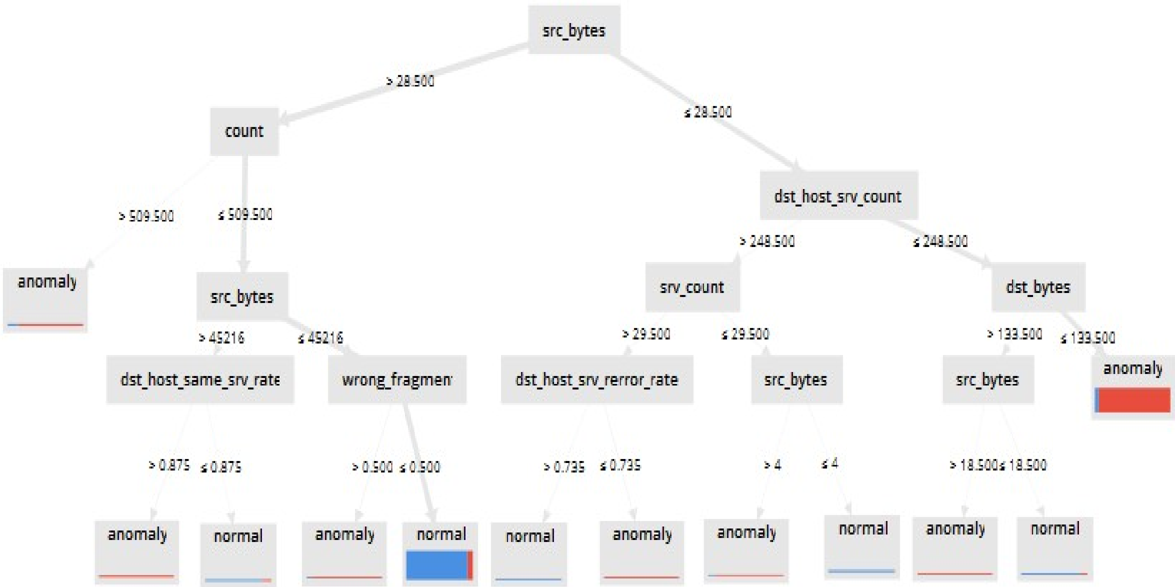

Fig. 3. C5.0 decision tree diagram on the NSL-KDD data set.

An important preference of the C5.0 decision tree model for testing features is the gain ratio. A higher gain ratio indicates a better model [31], [33]. The gain ratio is calculated as below:

$$\text{(1)} \qquad \text{Gain ratio}\ (K, C) = {\text{Gain}\ (K, C)}\Big/{\text{Split Info}\ (K, C)}$$

According to Equation 1, $Gain\ (K, C)$ and $Split\ Info\ (K, C)$ are calculated as follows:

$$\text{(2)} \qquad \text{Split Info}(K, C) = \text{Info}_{\text{entropy}}(|C1|/|C|, |C2|/|C|, \ldots, |Ci|/|C|)$$

$$\text{(3)} \qquad \text{Gain}\ (K, C) = \text{Info}_{\text{Entropy}}(K) - \text{Info}_{\text{Gain}}(K, C)$$

where K is the count of the features and C_i is the partition of the C derived by the value of K.
Thus, $Info_{entropy}\ (K)$ and $Info_{gain}\ (K, C)$ are formulated as follows:

$$\text{(4)} \qquad \text{Info}_{\text{Entropy}}(K) = -\sum_{j=1}^{N \in C_i} P_i \log_2 P_i$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

(5) $$\mathrm{Info}_{\mathrm{Gain}}(\mathrm{K},\mathrm{C}) = \sum_{\mathrm{i}=1}^{\mathrm{N}\in\mathrm{C_i}} \mathrm{P_i} \times \mathrm{Info}_{\mathrm{Entropy}}(\mathrm{K_i})$$

According to Equations 4 and 5, *P* is calculated as follows:

(6) $$\mathrm{P} = (|\mathrm{C1}|/|\mathrm{S}|, |\mathrm{C2}|/|\mathrm{S}|, \dots, |\mathrm{Ci}|/|\mathrm{S}|)$$

where $|S|$ is the count of examples in set of *S* and *P* is the probability distribution of partition (*C1*, *C2*, …, *Ci*).

3.4.2. Bayesian network model

Another classification model in data mining is Bayesian network classification model. The philosophy of this model is based on a possible framework for solving classification problems. According to Bayes' theorem, the classification of events is formed based on the probability of occurring or not occurring an event so that the probability of an event is calculated and classified [33]. In the Bayes' theorem, we have the following probabilities:

(7) $$P\,(D|B) = {P\,(B,D)}\big/{P(B)}$$

(8) $$P\,(D|B) = {P\,(B,D)P(D)}\big/{P(B)}$$

In fact, the Bayesian Network model has a graphical scheme that represents prediction variables and their eventual connections using a directed or non-circular signal graph. The nodes are also a prediction variable in the graph [33], [34].

3.4.3. Deep learning neural network model

The philosophy of deep learning is derived from the architecture of biological neural networks in human brain under artificial neural networks, which is a branch of machine learning and artificial intelligence. In nerve cells and neurons, information and data are in the form of pulses or electrical signals that enter and leave the cell. In other words, nerve cells decode through tagging and assigning features and items to different categories and classes so that a series of changes and processing are performed on the cell nucleus. These changes and processes are learned during human life, and the so-called neural network structure is trained during human life. A similar process is seen in deep learning neural network [35], [36], [37], [38], [39], [40], [41], [42]. In deep learning, we deal with multi-layered deep neural networks, which introduce multi-layered learning of the features as the main characteristic. These layers are called hidden layers in the neural network, and a network is considered as a deep learning network, when it includes more than two hidden layers. In general, this model has 3 types of layers:

- Input layer: Receives input data related to features.
- Hidden layer: Data patterns are extracted in this layer.
- Output layer: Data processing results are related to this layer.

It is necessary to mention that the advantage of a deep neural network is having lots of hidden layers, which makes it different from superficial artificial neural network that has a single hidden layer. This means that deep neural network is able to do more complex tasks. The structure of a deep network is such that the data is transferred from one hidden layer to another so that simpler features are recombined and recomposed as complex features.

For example, consider a two-layered neural network in which a three-dimensional input can be connected to four other neurons of different weights in one layer. This process is similar to feature extraction process, i.e. an input from a 3-dimensional space is mapped to a new 4-dimensional space, which can be known as the feature space. In other words, the inputs are transformed to a series of features that are good and useful features. In machine learning, after the feature extraction process, we have algorithm learning process, i.e. the features are used as inputs to a classification algorithm, which learns to detect the class of the inputs. We had the same rules and principles for these methods in Subsections 3.4.1 and 3.4.2. In this two-layer neural network, there is a layer called the feature layer or hidden layer, the outputs of which are the feature space and also the inputs to the last layer. The last layer is called the classification layer, which specifies the class of the input data related to the features. The two-layer neural network is shown in Fig. 4.

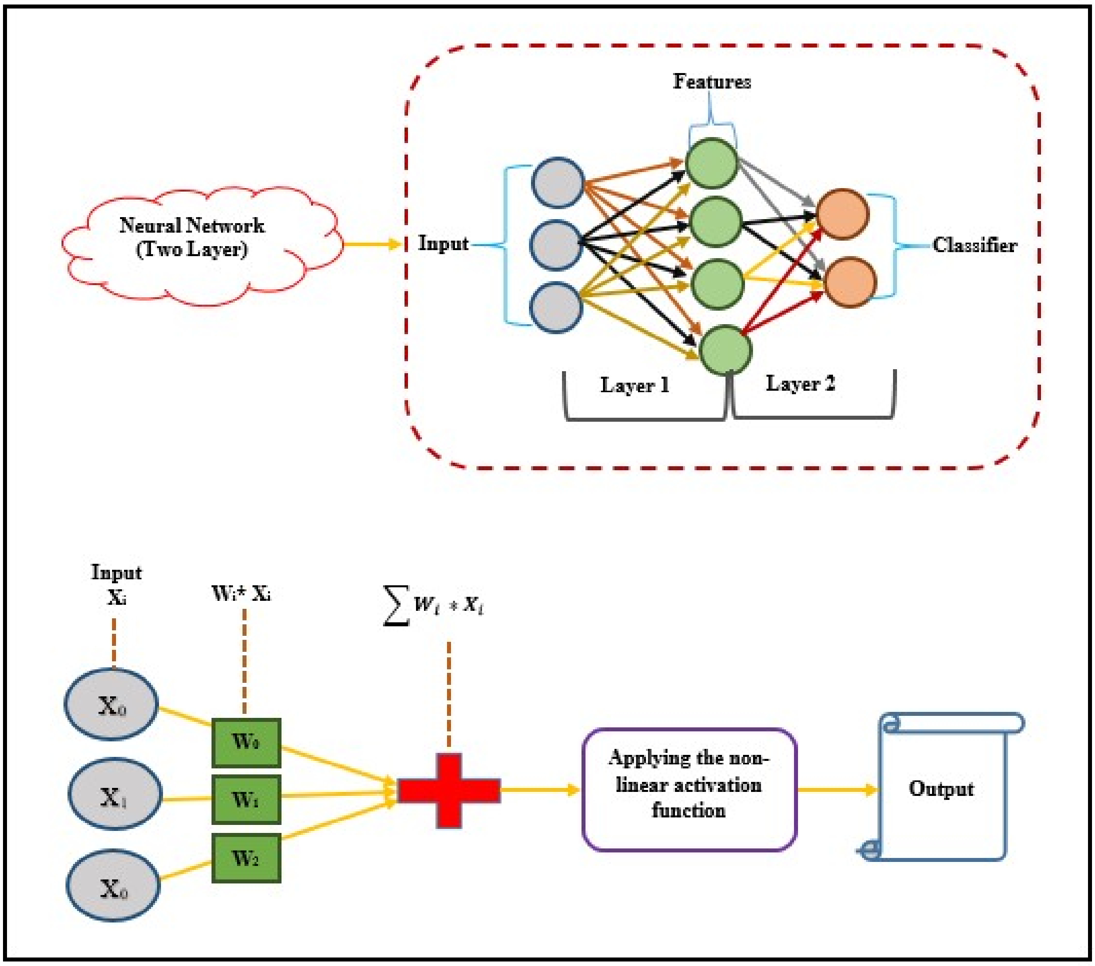

Fig. 4. Multi-layered neural network [33], [36].

According to Fig. 4, the output generation process is such that if we multiply each of the input dimensions by a coefficient or so-called weight, and then pass sum of them through a nonlinear function, a new output is generated. This is similar to the process that we have in a nerve cell, meaning that the input changes during the passage through the cell. These changes are weights, and the nonlinear function results in new outputs. The set of inputs, weights and nonlinear function are called a layer in artificial neural networks, and this layer must be well trained. Neural network training means finding weights to transform inputs into expected outputs. Therefore, it's the weights that are trained, and the nonlinear function is usually added to increase the network capability.

As mentioned before, the deep learning model of a neural network includes more than two middle or hidden layers. For example, in Fig. 5, a deep network with 3 hidden layers [43] using RapidMiner tool is illustrated. In a 3-layer network, low-level, middle-level and high-level or much more complex features are extracted in the first, second and third layers, respectively. At the output of this network, we have the classification of the input data that specifies the class type of the data.

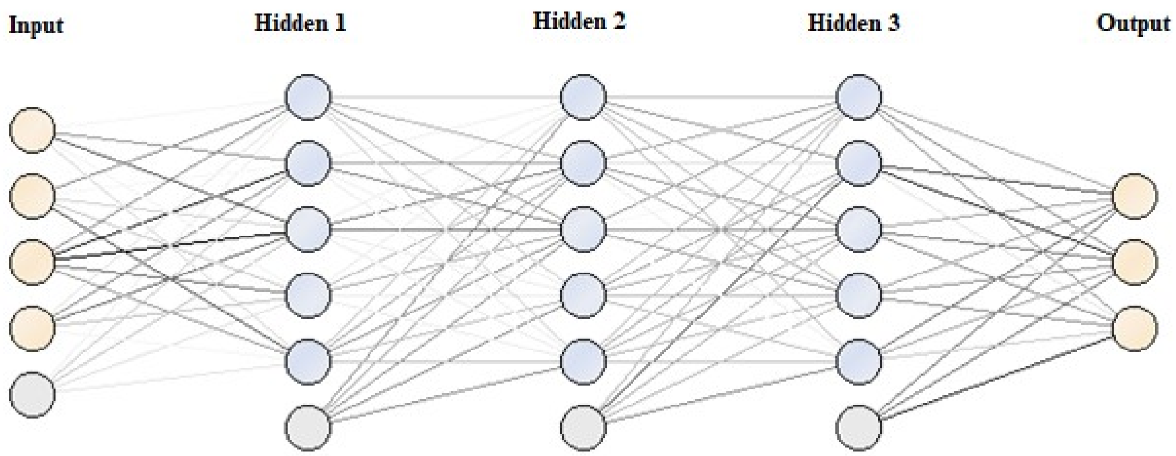

Fig. 5. Neural network with 3 hidden layers.

Hence, the goal of deep learning is to discover several levels of distributed representations of the input data so that by creating features in the lower layers, it can differentiate the factors of changes in the input data and then combine these representations in the higher layers [44]. In addition, one of the noticeable advantages of a deep network is that deep learning model performs very well on unstructured data and has a higher accuracy comparing to machine learning models such as decision tree, Bayesian network, support vector machine, etc., but in practice, requires a large amount of training data along with appropriate hardware and software. Furthermore, one of the most important capabilities of a deep learning network is the ability to extract features automatically. Deep neural network also has a high generalizability, meaning that in addition to the data being trained, if the network receives new data that is similar to the training data, it can detect the data with high accuracy, which is called high generalization ability. In this paper, a 6-layer deep learning model with 4 hidden layers sized 50x50 in the 10 epoch range by 10-fold cross validation method is used. Additionally, the nonlinear activation function used, which determines the activity of neurons in the hidden layers of the network, is determined by Maxout [45]. The Maxout function selects the maximum coordinates for the network input vector, and is utilized in this research to avoid data over-fitting and improve network training. The Sofmax function is also used to classify the output layer. The deep learning model used in this paper is performed in RapidMiner software. The proposed deep learning model is shown in Fig. 6.

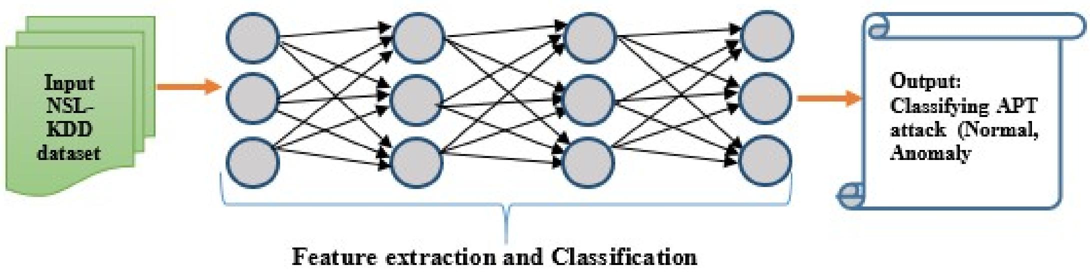

Fig. 6. Proposed deep learning model.

According to Fig. 6, after entering the data into the deep network, the extraction of features and classification of attacks is performed in combination and simultaneously, and no other method is required to extract the features, because feature extraction is performed automatically in deep network. Finally, the attack classification is accomplished after applying the nonlinear function.

4. Method Evaluation

In this paper, the confusion matrix is used to evaluate the proposed models [31], [32], [33], [46]. This matrix includes 4 elements, including True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The basic definitions of these 4 elements are as follows.

- TP: Represents that when an alert is generated, then an APT attack occurs.
- FP: Represents that when an alert is generated, but an APT attack does not occur.
- TN: Represents that when an alarm is not generated, then an APT attack does not occur.
- FN: Represents that when an alert is not generated, but an APT attack occurs.

The confusion matrix is shown in Table 1.

Table 1. Confusion matrix for detection of APT attack.

The Actual class	The predicted class	
	Anomaly	Normal
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Consequently, according to the confusion matrix, we have used 7 criteria to evaluate three models including Bayesian, C5.0 decision tree and deep learning. The criteria are accuracy, F-measure or F1-score, precision or positive predictive value (PPV), specificity (TNR), sensitivity or true positive rate (TPR), FPR and ROC-AUC score [33].
These criteria are formulated based on the following equations:
(9) $\text{Specificity} = \text{True Negative Rate (TNR)} = \text{TN} / \text{TN} + \text{FP}$
(10) $TPR = TP / TP + FN$
(11) $Accuracy = TP + TN / TP + TN + FP + FN$
(12) $Precision = TP / TP + FP$
(13) $F - measure = 2 * Precision * Recall / Precision + Recall$
Furthermore, FPR and FNR criteria show the type of false, and FPR is a more important criterion than FNR in terms of false determination and effectiveness. These criteria are formulated as follows:
(14)-(15)
These criteria are calculated through 10-fold cross validation method. The results of the classification models will be analyzed in the next Section.
(14) $\text{FPR} = 1 - \text{Specificity}$
(15) $FNR = 1 - TPR$

5. Experiment Results

The results of the Bayesian, C5.0 decision tree and deep learning classification models are investigated and analyzed in this Section. Since the purpose of this article is detection and classification of APT attacks on network, we have used the NSL-KDD data set to detect the APT attacks. In the detection process, after receiving the data and pre-processing, we have used classification models such as Bayesian, C5.0 decision tree and 6-layer deep learning. In addition, to evaluate the models in the output, criteria such as accuracy, precision, false positive rate (FPR), false negative rate (FNR), sensitivity, specificity and F-measure have been extracted as experiment results. In this experiment, 10-fold cross validation method has been utilized to classify the data set so that in each fold, 0.9 of the data has been used for training and the remaining 0.1 of the data has been used to test the performance of the proposed models, and this process has been repeated 10 times. Moreover, in order to evaluate the generated models, control the training process, prevent over-fit of the data and improve the generalization, we have considered 90% of the training data, 80% of which is for training and 20% is for validation of the models in 10 epochs. Fig. 7 illustrates the training, testing and validation process of the proposed models.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

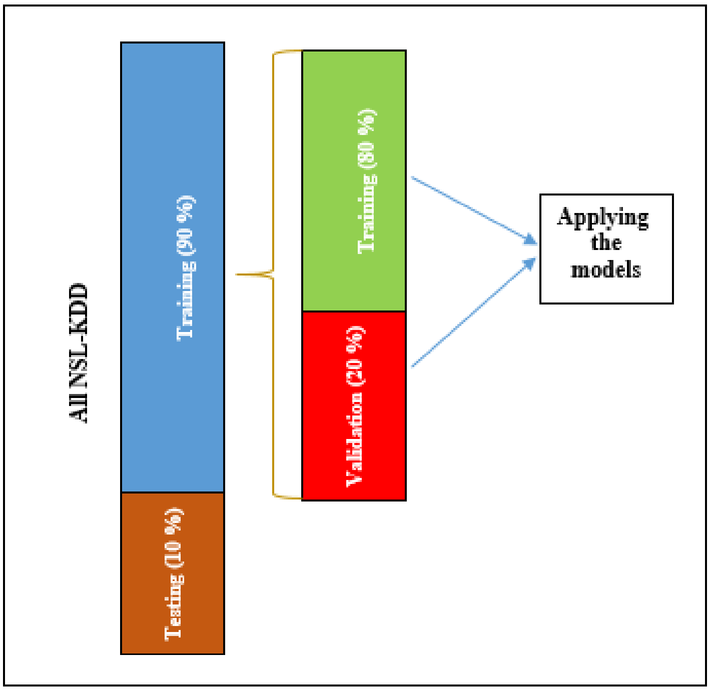

Fig. 7. Training, testing and validation process of the proposed models.

The results based on the evaluation criteria are given in Table 2 for the Bayesian, C5.0 decision tree and deep learning classification models.

Table 2. Results of the classification models (%).

Classification models	ACC	TPR	TNR	PPV	F-measure	FPR	FNR
Naïve Bayes	88.37	87.12	89.53	88.54	87.82	10.47	12.88
Decision Tree of C5.0	95.64	97.30	97.44	97.15	95.39	2.56	2.70
Deep Neural Network (DNN)	98.85	98.89	98.87	98.72	95.84	1.13	1.11

According to Table 2, the accuracy of the Bayesian network, C5.0 decision tree and deep neural network classification models is 88.37%, 95.64% and 98.85%, respectively. Besides, for the important FPR criteria, the values 1.13, 2.56 and 10.47 are obtained for the deep neural network, C5.0 decision tree and Bayesian network, respectively. Furthermore, for the rest of the evaluation criteria, the proposed deep learning model has achieved the best results.
In addition to the above criteria, in terms of TPR, TNR, F-measure and FNR criteria, the 6-layer deep learning model performs better than the C5.0 decision tree and Bayesian network models.
Another important criterion used in this experiment is the AUC criterion, the accuracy of the surface below the ROC diagram. The better AUC value indicates the more accuracy of the model. The diagram of this criterion for the Bayesian network, C5.0 decision tree and deep learning classification models are shown in Figs. 8-10, respectively.

 doi:10.20944/preprints202007.0745.v1

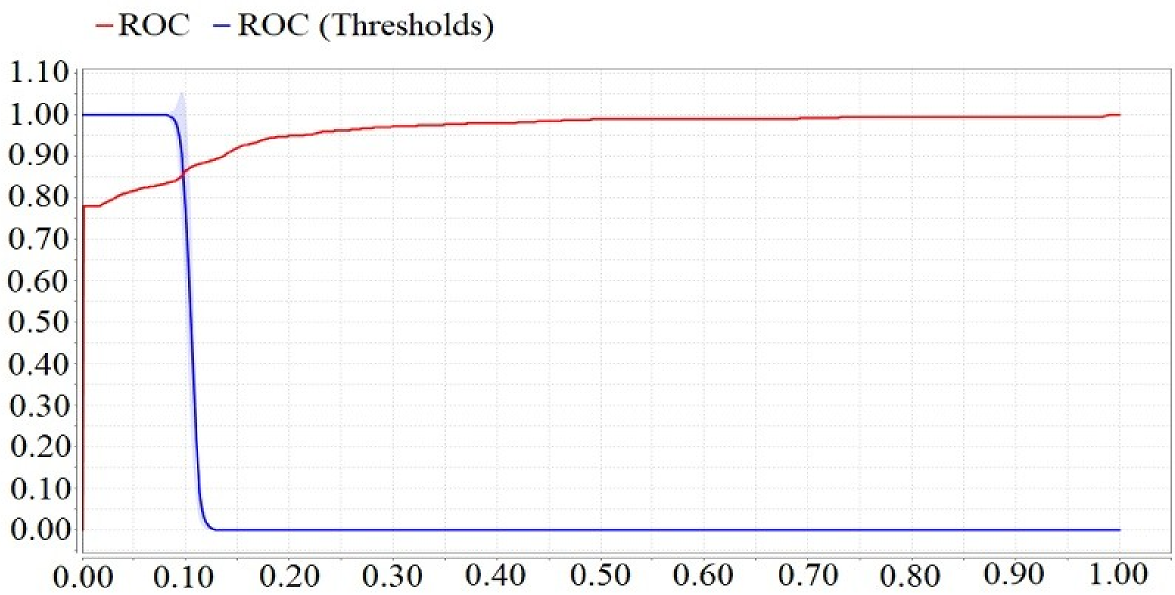

Fig. 8. ROC curve for the Bayesian model.

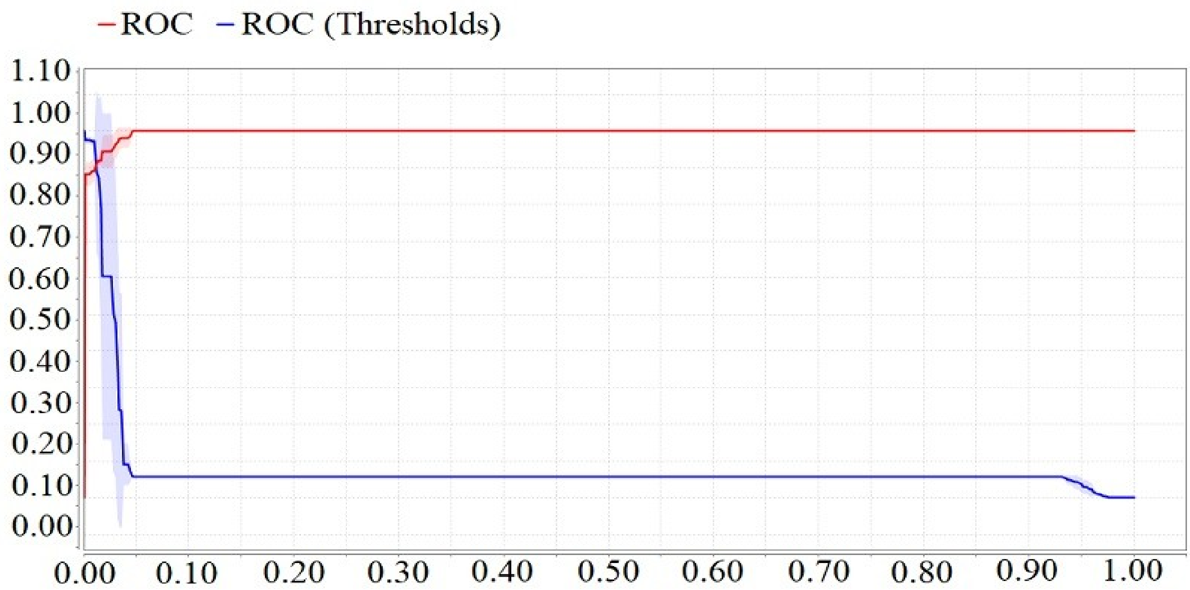

Fig. 9. ROC curve for the C5.0 decision tree model.

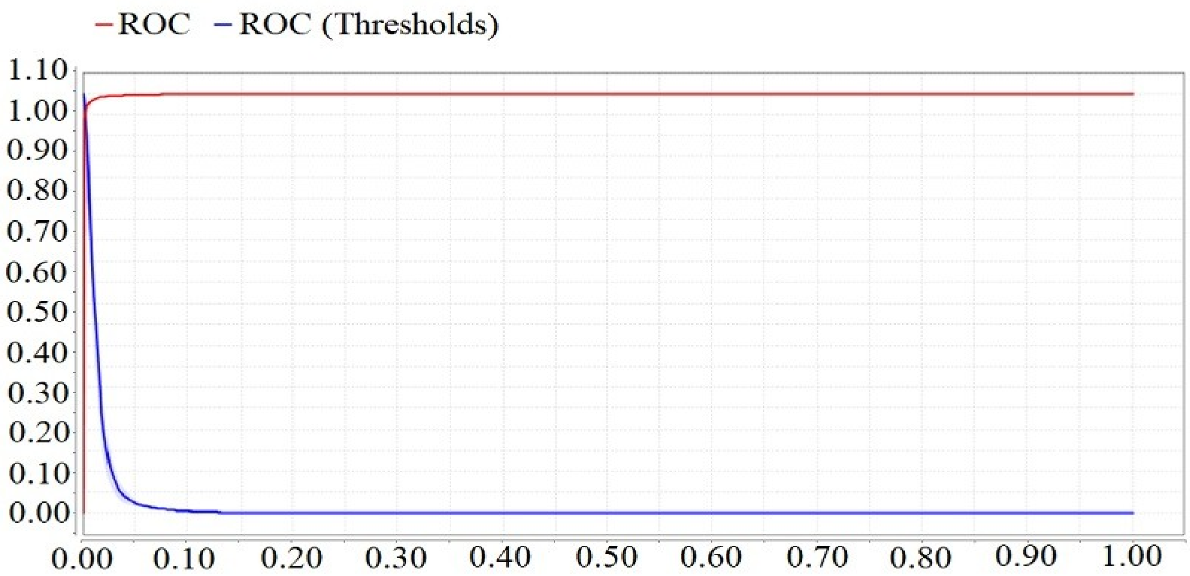

Fig. 10. ROC curve for the 6-layer deep learning model.

According to Figs. 8-10, the AUC value for the Bayesian, C5.0 decision tree and deep learning classification models are obtained as 96.1%, 99.60% and 99.90%, respectively. Consequently, the proposed deep learning model is the best model in terms of the AUC. As a result, by analyzing the classification models, it can be concluded that the 6-layer deep learning model has the best performance regarding all the criteria examined in the output to detect APT attacks on the NSL-KDD data set. Since the purpose of this paper is to extract features automatically in the layers related to the features using a deep neural network, the important features that are extracted using this method are explained in Fig. 11.

```
Variable Importances:
                Variable Relative Importance Scaled Importance Percentage
               srv_count            1.000000          1.000000   0.012395
           diff_srv_rate            0.982102          0.982102   0.012173
           service.shell            0.982063          0.982063   0.012173
         service.daytime            0.975109          0.975109   0.012086
           service.eco_i            0.972955          0.972955   0.012060
             rerror_rate            0.954541          0.954541   0.011832
      dst_host_srv_count            0.940606          0.940606   0.011659
      service.netbios_ns            0.924339          0.924339   0.011457
             service.X11            0.914025          0.914025   0.011329
         srv_rerror_rate            0.907292          0.907292   0.011246
---
         service.netstat            0.326361          0.326361   0.004045
            service.nnsp            0.325659          0.325659   0.004037
      protocol_type.tcp             0.323045          0.323045   0.004004
     service.netbios_ssn            0.321485          0.321485   0.003985
                flag.OTH            0.315218          0.315218   0.003907
                 flag.S3            0.286562          0.286562   0.003552
           service.pop_2            0.257605          0.257605   0.003193
```

Fig. 11. Features extraction using a deep neural network on the NSL-KDD data set.

According to Fig. 11, the most important features/variables of the APT attack detection are arranged from the highest probability of occurrence to the lowest probability so that the highest probability of attack for srv_count variable, which indicates the count of connections to the identical services with the current connection during the last two seconds [30], is 1.0000.

6. Results and Discussion

In general, researches indicate that among the approaches developed for APT attacks detection, artificial intelligence methods are the best methods. Moreover, according to the latest scientific achievements in the field of network security, deep learning method has had the best performance comparing to other methods. Consequently, in this paper, artificial intelligence methods such as C5.0 decision tree, Bayesian network and deep neural network classification models were used to detect two normal and anomaly classes of APT attacks on the NSL-KDD data set. The models were implemented via RapidMiner software. As APT attack is one of the most stable and persistent attacks on the system and involves the system for a long time, it is very important to detect it early. Therefore, we needed artificial intelligence methods for timely detection of APT attacks, and we implemented three methods of C5.0 decision tree, Bayesian model and deep learning using 10-fold cross validation method. According to Table 2, by evaluating the criteria, we concluded that the accuracy of the deep learning model reaching 98.85% is the best compared to the C5.0 decision tree and Bayesian models reaching 95.64% and 88.37%, respectively. Another important criterion is the FPR criterion, which is 1.13, 2.56, and 10.47 for the deep learning network, C5.0 decision tree and Bayesian network models, respectively. In addition, for the rest of the evaluation criteria, including TPR, TNR, F-measure and FNR, the deep learning model performed better than the C5.0 decision tree and Bayesian network models.
In addition, in terms of the AUC criterion, according to Figs. 8-10, the AUC value for the Bayesian, C5.0 decision tree and deep learning models is obtained as 96.1%, 99.60% and 99.90%, respectively. Thus, regarding AUC, deep learning model is a more appropriate model for detection and classification of APT attacks. Comparison between the C5.0 decision tree, Bayesian Network and deep learning classification models in terms of the AUC criterion via ROC curve is illustrated in Fig. 12 and comparison between the models based on the mentioned criteria are shown in Fig.13.

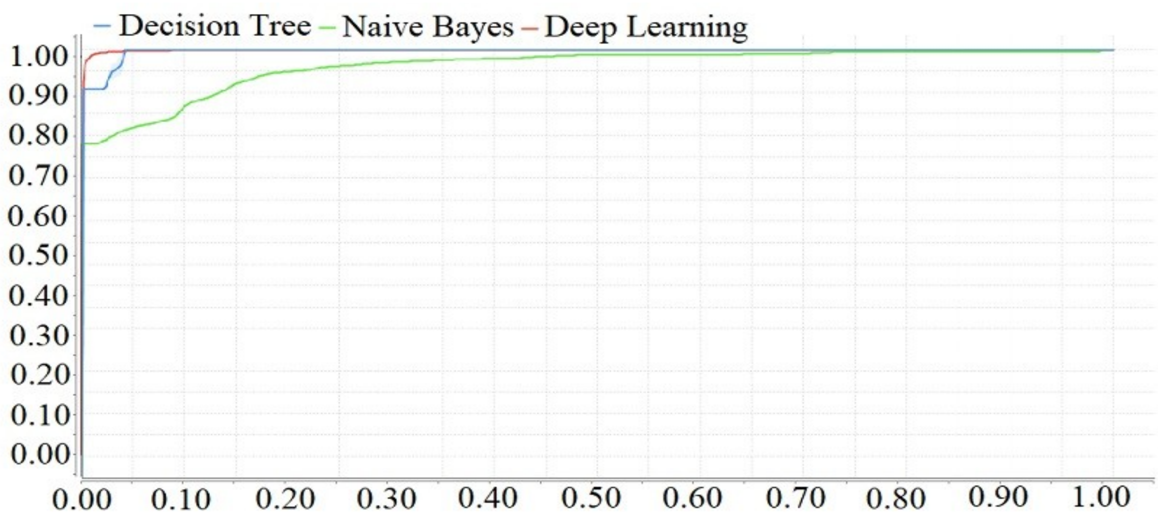

Fig. 12. Comparison between the C5.0 decision tree, Bayesian Network and deep learning classification models in terms of the AUC criterion via ROC curve.

 doi:10.20944/preprints202007.0745.v1

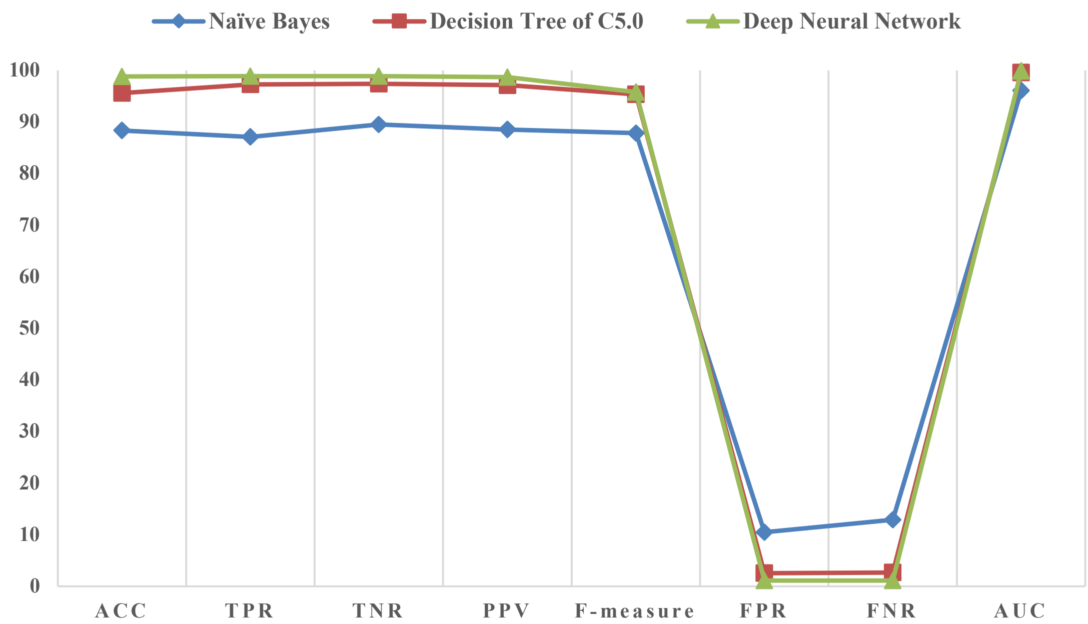

Fig. 13. Comparison between the classification models based on the evaluation criteria.

According to Fig. 13, comparison between the models show that the deep learning model provides the best performance for detection and classification of APT attacks regarding ACC, TPR, TNR, PPV, F-measure, FPR, FNR and AUC criteria.

Finally, based on the results obtained, we can conclude that the deep learning model has been selected as the proposed model of this paper. As an acceptable result for the proposed deep learning model, we have implemented Lift chart [33] diagram for two normal and anomalous classes with a confidence index on the test data set. In Figs. 14 and 15, Lift chart diagram is shown for normal and anomalous classes based on the APT attack detection, respectively.

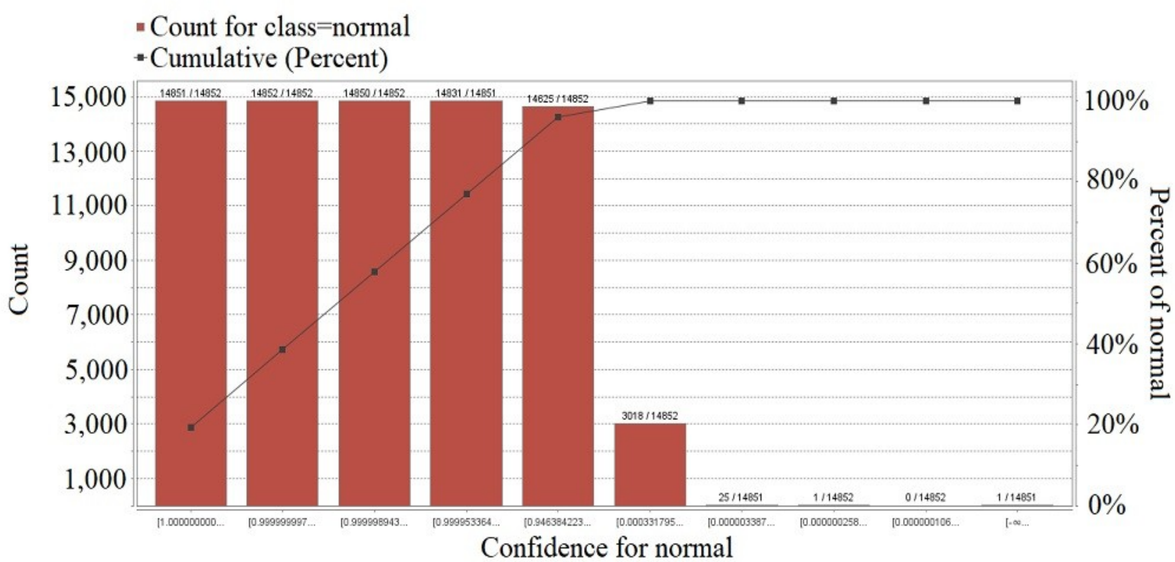

Fig. 14 Lift Chart diagram for modeling of Deep Neural Network for normal class.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

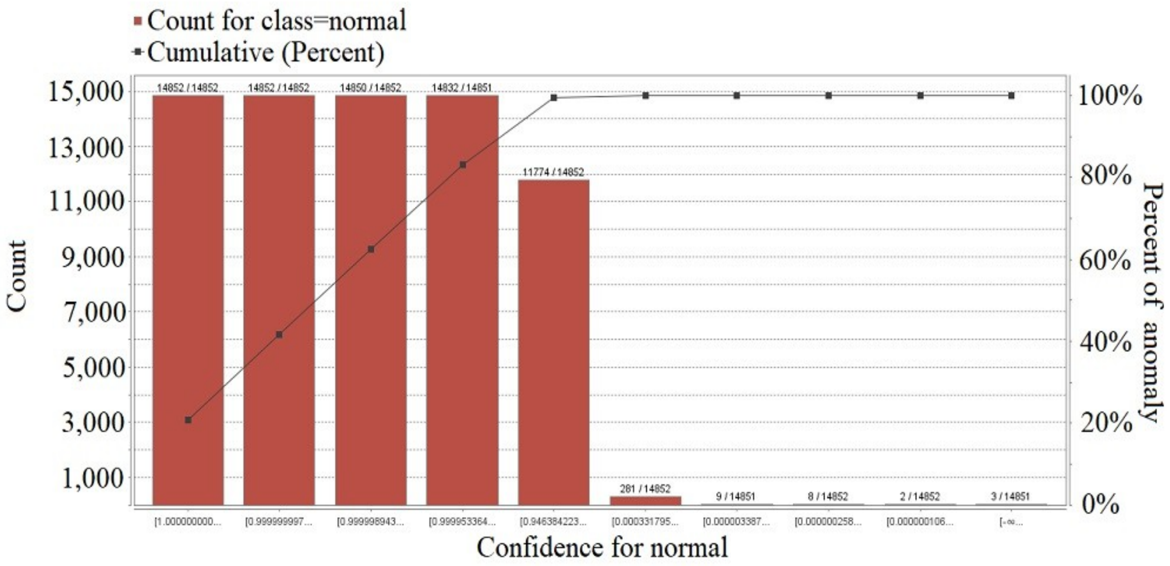

Fig. 15 Lift Chart diagram for modeling of Deep Neural Network for anomaly class.

The diagram of Lift Chart for normal class is illustrated in Fig. 14. Based on Fig. 14, confidence for normal class, for example in scope 0.94 related to the fifth record include 14851 samples, illustrating that 14625 samples are as normal. Therefore, confidence for normal class at the scope 0.94 is very important for 14851 records, illustrating that over 98% of samples are as normal. Furthermore, the diagram of Lift Chart for anomaly class is shown in Fig. 15. Hence, confidence for anomaly class, for example in scope 0.99 related to the fourth record contains 14851 samples illustrating that 14832 samples are as anomaly. Therefore, confidence for anomaly class on data sets at the scope 0.99 is very important for 14851 records, illustrating that more than 98% of samples are as anomaly.
Eventually, a comparison between deep neural network model with other works in terms of the accuracy and FPR obtained on the NSL-KDD data set is demonstrated in Table 3.

Table 3. Comparison between the deep learning model and previous works in terms of the FPR and accuracy.

Authors	**Dataset used**	**Technique used**	**FPR**	**ACC**
Ghafir et al. [8]	Network Traffic	linear SVM	Not considered	84.8%
Chu et al. [9]	NSL-KDD	SVM-RBF	Not considered	97.22%
Marchetti et al. [15]	Network Traffic	Framework	Not considered	Not considered
Bodström et al. [14]	Network Traffic	Deep Learning stack by exploiting sequential neural networks	Not considered	Not considered
Ghafir et al. [10]	Network Traffic	Hidden Markov Model	Not considered	91.80%
Bodström and Hämäläinen, [26]	Network Traffic	OODA LOOP	Not considered	Not considered
Proposed	NSL-KDD	6-layer deep learning model	1.13	98.85

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 July 2020 doi:10.20944/preprints202007.0745.v1

According to Table 3, we have accomplished the APT attack detection and classification process on the NSL-KDD data set and have compared our proposed method with the work studied in [9]. Chu et al. [9] have used an SVM classifier with an RBF kernel in order to detect the APT attack on the NSL-KDD data set, and their evaluation criterion regarding accuracy has reached 97.22%, while they have not considered the FPR criterion. However, in this study, we have utilized a 6-layer deep learning model to achieve a detection accuracy of 98.85%, and considering the FPR evaluation criterion, its value has been 1.13. Consequently, the proposed deep learning method has the best performance in comparison with the previous work providing the advantage of considering the FPR criterion for the first time.

7. Conclusion and Future Works

In this study, three artificial intelligence-based classification models including Bayesian Network, C5.0 decision tree and deep learning were used to detect and classify APT attacks on the NSL-KDD data set. Since the nature of the APT attack is permanent and persistent presence in the victim system, early detection of this attack requires high accuracy and minimal FPR in the early stages. For this purpose, through the mentioned classification models, based on the obtained results, a 6-layer deep learning model with the highest accuracy and the lowest FPR, which are equal to 98.85 and 1.13, respectively, was selected as the final model. In addition, other evaluation criteria, such as TPR, TNR, PPV, F-measure, FPR, FNR and AUC were investigated. The 6-layer deep learning model had also the best performance in terms of these criteria. One of the important criteria for comparing models is the AUC criterion. Figs. 8-10 as well as Fig. 12, comparing the three classification models, show that the deep learning model with the AUC value 99.9% is better than the Bayesian Network and C5.0 decision tree models with the AUC values 99.6% and 99.60%, respectively. Finally, a comparison table was made to compare the proposed deep learning model with other related work. As shown in Table 3, the 6-layer deep learning model had the best execution and performance in terms of the accuracy compared to previous work [9] regarding APT attack detection on the NSL-KDD data set. Furthermore, so far in no study the important features of the data set have been extracted. Fig. 11 shows the importance of the features. As an important result, deep learning has been ranked the highest and best in most areas of network security detection, and in this article, we also have obtained the best results for the deep learning model. For future work, we suggest that a combination of machine learning and deep learning methods can be implemented on the NSL-KDD data set used and network traffic flow. Moreover, supervised and unsupervised deep learning methods, such as Recurrent Neural Networks and Auto-Encoder Neural Networks, respectively, can be utilized.

References

[1]. Y. Wang, Q. Li, Z. Chen, P. Zhang, and G. Zhang, "A Survey of Exploitation Techniques and Defenses for Program Data Attacks," *Journal of Network and Computer Applications*, vol. 154, p.102534, 2020.
[2]. J. Chen, C. Su, K.H. Yeh, and M. Yung, "Special issue on advanced persistent threat," *Future Generation Computer Systems*, vol. 79, pp. 243-246, 2018.
[3]. S. Singh, P. K. Sharma, S. Y. Moon, D. Moon, and J. H. Park, "A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4543-4574, 2019.
[4]. A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no, 2, pp. 1851-1877, 2019.
[5]. M. Auty, "Anatomy of an advanced persistent threat," *Network Security*, vol. 2015, no. 4, pp. 13-16, 2015.
[6]. https://malpedia.caad.fkie.fraunhofer.de/actor/turla_group
[7]. https://www.cynet.com/cyber-attacks/advanced-persistent-threat-apt-attacks/
[8]. I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, pp. 349-359, 2018.
[9]. W. L. Chu, C. J. Lin, and K. N. Chang, "Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine," *Applied Sciences*, vol. 9, no. 21, p.4579, 2019.
[10]. I. Ghafir, K. G. Kyriakopoulos, S. Lambotharan, F. J. Aparicio-Navarro, B. AsSadhan, H. BinSalleeh, and D. M. Diab, "Hidden Markov models and alert correlations for the prediction of advanced persistent threats" *IEEE Access*, vol. 7, pp. 99508-99520, 2019.
[11]. M. Lee and D. Lewis, "Clustering disparate attacks: Mapping the activities of the advanced persistent threat," *Virus Bulletin Conference October 2011*, *Last accessed June* 26, 2013.
[12]. B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "Duqu: Analysis, detection, and lessons learned," In *ACM European Workshop on System Security (EuroSec)*, Vol. 2012, 2012.
[13]. M. Balduzzi, V. Ciangaglini, and R. McArdle, "Targeted attacks detection with spunge," In *IEEE Conference on Privacy, Security and Trust*, pp. 185-194, 2013.
[14]. T. Bodström and T. Hämäläinen, "A novel deep learning stack for APT detection" *Applied Sciences*, vol. 9, no. 6, p.1055, 2019.
[15]. M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Computer Networks*, vol. 109, pp. 127-141, 2016.
[16]. P. Bhatt, E. T. Yano, and P. Gustavsson, "Towards a framework to detect multi-stage advanced persistent threats attacks," In *IEEE international symposium on service oriented system engineering*, pp. 390-395, 2014.
[17]. R. Brewer, "Advanced persistent threats: minimising the damage," *Network security*, vol. 2014, no. 4, pp. 5-9, 2014.
[18]. N. Virvilis and D. Gritzalis, "The big four-what we did wrong in advanced persistent threat detection?," In *IEEE international conference on availability, reliability and security*, pp. 248-254, 2013.
[19]. B. Bencsáth, G. Pék, L. Buttyán, and M. Felegyhazi, "The cousins of stuxnet: Duqu, flame, and gauss," *Future Internet*, vol. 4, no. 4, pp. 971-1003, 2012.
[20]. T. M. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91-93, 2011.
[21]. M. H. Au, K. Liang, J. K. Liu, R. Lu, and J. Ning, "Privacy-preserving personal data operation on mobile cloud—Chances and challenges over advanced persistent threat," *Future Generation Computer Systems*, vol. 79, pp. 337-349, 2018.
[22]. B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE network*, vol. 8, no. 3, pp. 26-41, 1994.
[23]. M. Roesch, "Snort: Lightweight intrusion detection for networks," *Lisa*, vol. 99, no. 1, pp. 229-238, 1999.
[24]. D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, vol. SE-13, no. 2, pp. 222-232, 1987.
[25]. J.R. Johnson and E. A. Hogan, "A graph analytic metric for mitigating advanced persistent threat," In *IEEE International Conference on Intelligence and Security Informatics*, pp. 129-133), 2013.
[26]. T. Bodström and T. Hämäläinen, "A Novel Method for Detecting APT Attacks by Using OODA Loop and Black Swan Theory" In *Springer International Conference on Computational Social Networks*, pp. 498-509), Cham. 2018.

[27]. I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Computers & Security*, vol. 48, pp. 35-57, 2015.
[28]. https://github.com/jmnwong/NSL-KDD-Dataset
[29]. M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p.100059, 2019.
[30]. L. Dhanabal and 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.446-452, 2015.
[31]. J. H. Joloudari, E. Hassannataj Joloudari, H. Saadatfar, M. GhasemiGol, S. M. Razavi, A. Mosavi, N. Nabipour, S. Shamshirband, and L. Nadai, "Coronary artery disease diagnosis; ranking the significant features using a random trees model," *International journal of environmental research and public health*, vol. 17, no. 3, p.731, 2020.
[32]. J. Hassannataj Joloudari, E. Hassannataj Joloudari, H. Saadatfar, M. GhasemiGol, S. M. Razavi, A. Mosavi, N. Nabipour, S. Shamshirband, and L. Nadai, "Coronary Artery Disease Diagnosis; Ranking the Significant Features Using Random Trees Model," *arXiv*, pp. arXiv-2001, 2020.
[33]. J. H. Joloudari, H. Saadatfar, A. Dehzangi, and S. Shamshirband, "Computer-aided decision-making for predicting liver disease using PSO-based optimized SVM with feature selection," *Informatics in Medicine Unlocked*, vol. 17, p. 100255, 2019.
[34]. J. Cheng and R. Greiner, "Learning bayesian belief network classifiers: Algorithms and system," In *Springer Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 141-151, Berlin, Heidelberg, 2001.
[35]. I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *MIT press*, 2016.
[36]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436-444, 2015.
[37]. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning,"*ICML*, 2011.
[38]. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85-117, 2015.
[39]. L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197-387, 2014.
[40]. Y. Bengio, I. Goodfellow, and A. Courville, "Deep learning," *MIT press*, vol. 1, 2017.
[41]. M. A. Nielsen, "Neural networks and deep learning," *Determination press*, vol. 2018, 2015.
[42]. Y. Bengio, "Learning deep architectures for AI," *Now Publishers Inc*, 2009.
[43]. Q. Zhu, X. Jiang, Q. Zhu, M. Pan, and T. He, "Graph embedding deep learning guide microbial biomarkers' identification," *Frontiers in Genetics*, vol. 10, p.1182, 2019.
[44]. M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11-24, 2014.
[45]. I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," In *International conference on machine learning*, pp. 1319-1327, 2013.
[46]. S. Mojrian, G. Pinter, J. H. Joloudari, I. Felde, N. Nabipour, L. Nadai, and A. Mosavi, "Hybrid Machine Learning Model of Extreme Learning Machine Radial basis function for Breast Cancer Detection and Diagnosis; a Multilayer Fuzzy Expert System," *arXiv preprint arXiv*: 1910.13574, 2019.