Network Intrusion Detection Using Supervised & Un–Supervised Machine Learning Algorithms

Mohammed Sufiyan Saqib
Department of Information Science & Engineering, Software Engineering
Ramaiah Institute of Technology, Bengaluru, Karnataka, India

Abstract— In recent years, with the enormous volumes of data produced every day because of the rapid increase in computer/mobile usage and digital computer network, the defense of the computer system becomes more and more crucial. This brings a growing concern in information security and analysis of data, as network hackers discover new network intrusion attacks day by day. Network Intrusion detection system prevents network attacks by monitoring the computer network and analyzing the data in the computer system or the computer network. There have been various researches conducted to find an efficient solution to prevent network intrusion to ensure privacy and security in the computer network. But with the emergence of high volume, variety, and velocity of data in the computer network, the traditional techniques for network intrusion detection face difficulty in the data analysis process to detect attacks. Machine learning techniques prove to be a very effective tool for analysis of abnormal behavior as it provides supervised and unsupervised methods for clustering classification and regression. These techniques can identify any anomalous behavior in computer network traffic. The principal aim of this paper is to develop a machine learning model which can work on a dataset with no output label i.e., no dependent variable and should be able to classify any abnormal behavior in the network traffic, to satisfy this aim a combination of the unsupervised and supervised machine learning algorithms is used to develop a hybrid machine learning model which will cluster and classify any abnormal behavior in network traffic. The CICIDS2017 dataset is used to train and test the proposed model. The algorithms used to develop the hybrid machine learning model are Principal Component Analysis/PCA (unsupervised machine learning) used for dimensionality reduction, Kohonen Self Organized Maps (unsupervised machine learning) used for visualization and clustering and Artificial Neural Networks (supervised Machine learning) used for regression and classification. The accuracy calculated for the evaluation of the developed hybrid model is 82%, with a False Positive of 9.40% and False Negative of 6.64%.

Keywords: Network Intrusion Detection System, Machine Learning, Principal Component Analysis, Kohonen Self Organized Maps, Artificial Neural Networks

I. INTRODUCTION

Network Security Maintenance is not only for handling network privacy issues or protecting data but also helps to avoid many hazardous situations in the network. This makes it a major safety concern for neutralizing such unwanted situations in today's world. The security intelligence report provided by Microsoft for January-June 2010 shows a high rate of increase in network infection trends [1].

Various researches have been conducted to find an efficient and powerful solution for the prevention of network intrusion in order to ensure privacy and security in the computer network. But with the emergence of high volume, variety, and velocity of data in the computer network, the traditional techniques for network intrusion detection face difficulty in the data analysis process to detect attacks.

Network Intrusion detection system prevents these attacks by monitoring the computer network and analysing the data in the computer system or the computer network. These network intrusion detections are of two types, which are anomaly-based detection and signature-based detection.

Anomaly-based detection is based on the analysis of the characteristics of a network, it possesses the capability for detecting anomaly behaviour by analysing the high volume of traffic, a flood of traffic from a particular host or to a particular host, imbalance of load in the network [2]. One of the main disadvantages of this type of network intrusion detection is that it fails to detect an anomaly when malicious behavior befalls inside the normal network. The advantage of this type of network intrusion detection is, a new attack with no signature can be detected if it behaves in a different way when compared with normal traffic behavior.

Signature-based detection is based on the analysis of a series of packets or bytes which are known to be an anomaly in the network traffic. Here for every attack, a signature is created, and it identifies only attacks with known signature, any attacks with an unknown signature are left undetected. One of the main disadvantages of this type of network intrusion detection is if a person knows the network behavior to be identified it becomes easy to understand and develop signatures.

Machine learning can be used as an effective tool as it provides supervised and unsupervised methods for classification. This classification technique can be used to identify any anomalous behavior in computer network traffic. Presently machine learning is being used extensively for implementing an effective and efficient network intrusion detection system. Machine learning algorithms such as neural network [3] , decision trees, support vector machines [4] seem to work efficiently significantly in network intrusion detection systems.

In this paper, a combination of supervised and un – supervised machine learning algorithms will be used to classify between normal and abnormal behavior in the network traffic.

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) has been widely used in many research fields and is a standard statistical method used in data analysis and pre-processing. PCA transforms

the data in a reduced form and to keep most of the original variance in the original data present [9]. PCA is used in a mathematical sense to transformation n correlated variables, called the principal components (PCs), into d (d<<n) uncorrelated variables [10].

Let's consider an M1 connection vector data set to be v11, v21, v31…., vM1 where each connection vector is represented by N1 functions. The stages to calculate the PCs are given below.

Stage 1: Calculate this set of data with an average μ

$$\mu = \frac{1}{M1}\sum_{j=1}^{M1} v_i$$

Stage 2: The mean deviation is defined as

$$\theta_j = v_j - \mu$$

Stage 3: Dataset sample covariance matrix

$$C1_{n*n} = \frac{1}{M1}\sum_{j=1}^{M1} \theta_j \theta_j^{T1} = \frac{1}{M1} A1A1^{T1}$$

Stage 4: Let $U1_k$ be the k^{th} eigenvector of C1, $\lambda 1_k$ be the eigenvalue and $U1_{n*d} = [U1_1, U1_2 .. U1_d]$ be the eigenvectors. Then we get

$$C1U1_k = \lambda 1_k U1_k$$

Stage 5: Order the eigenvalue in a lower order and pick the eigenvectors with the highest eigenvalues. The number of key components depends on the inertia ratio of:

$$\tau 1 = \frac{\sum_{j=1}^{n} \lambda 1_j}{\sum_{j=1}^{n} \lambda 1_j}$$

This ratio determines the amount of data saved from all the raw input data by the respective d equivalents.

Stage 6: Let t1 be a new vector for the sample column and project t1 into a new subsurface, spread over the rules of "PCi"

$$y_i = U1_j^T t1$$

III. KOHONEN SELF – ORGANIZING MAP

For the classification process neural networks are commonly used, the key neural networks classificatory for competitive learning is the Self-Organizing Map [11]. Competitive learning is a method that distributes the many learning input data in clusters [12].

The goal of the learning process is to locate the winning neurons that fit the input vector; they form the centers of the cluster. Neurons competing and the winners will have the Synaptic Weight Vector nearest to the input object vector. There are two layers, the input layer and the output layer of the Self-Organizing map (known as the Kohonen Network) [13]. The map elements dispersed throughout (usually 2D).

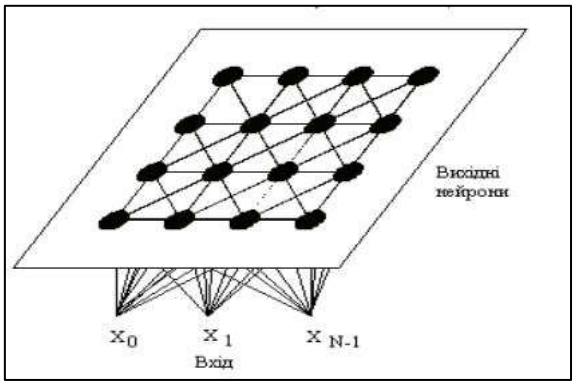

Fig. 1: Self - Organizing Map

The purpose of learning the card is to apply the following method of approximation. First, it allows random selection from the cluster centers available, and then the algorithm is charged to develop the cluster centers slowly so that the learning data are grouped together [14]. The main learning algorithm goes through a series of iterations successively, each processing the learning vectors independently. The network input will be submitted successively. The desired output vectors are not yet defined at this stage. Neurons similar to topology will react to similar input vectors afterwards. Compared to the input object, the winner and his neurons are modified [12].

$$w_{ij} = w_{ij}(t) + \alpha_i(t) h(d,t).[y_i - w_{ij}(t)]$$

y_i: Output Neuron i

$w_{ij}(t), w_{ij}(t+1)$: Synaptic Weights

α_i: Learning speed coeffecient

$$\alpha_i = \alpha_0 e_t^i$$

i: Iteration number

t: Iteration rate

$h(d,t)$: Neighbourhood function, defined as:

$$h(d,t) = \begin{cases} 0, & d \geq \delta(t) \\ e^{\left(\frac{d}{2\delta(t)}\right)}, & d < \delta(t) \end{cases}$$

$$\delta(t) = \delta_0 e^{\left(\frac{1}{\mu}\right)}$$

d: Distance between the neuron x and the winner neuron

δ_0: Constant

$$\mu = \frac{n}{\log_{10}(\delta_0)}$$

n: iteration rate

For all vectors the learning process will continue until the SOM stabilizes. The normalization for the set of inputs, as well as the synaptic weight values, is achieved during the learning process to reduce the learning process time [13]:

$$x_i = x_i \Big/ \sqrt{\sum_{j=0}^{n-1} x_j^2}$$

x_i: synaptic weight vector

n: object x length

During the analysis of the learning process, the characteristics of the Kohonen Network which influence the learning process and the results obtained were fixed [15]. The knowledge of the original vector languor will be lost after data normalization, and the SOM does not take into consideration absolute values of component artifacts, which affect classification accuracy when a linear dependency is detected between vectors [13].

When synaptic weights are induced by random values, the different classes are broadly distributed and excited or, if the subjects of this class are near, are segmented to subclasses. They are narrow. The convex combination method is used to prevent this problem [15]. The proximity (initial length of neuron topological area) is another significant parameter of this kind of network. The neurons spreading around the neighborhood will slowly be changed during the learning process, leading to an incremental decrease in the initial duration. The result of clustering is also based on the quantity and the number of neurons in the network output layer of the previously declared clusters. The reliability of the network is ensured by an invariant number of clusters, progressive weight changes and efficient learning processes. Learning speed and accuracy depends on the data size (vector length) and vector regularity [13].

IV. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network is based on human neural systems and is being used in various areas such as pattern recognition, optimization, control and so on. The neural network is made up of a number of processing units (nodes) with directed connections. The relationship of the input and output neurons is weighted [16].

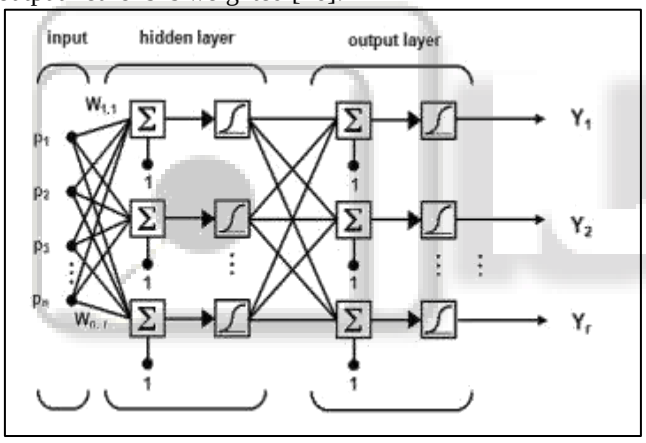

Fig0 2: Artificial Neural Network

A. *Feed – Forward Neural Network*

There are multiple neurons organized into layers such as input layers, hidden layers and output layers of the Multilayer Feed-Forward neural network (Fig.1). One or more neurons output layer produces one or more input outputs. The training process task in a neuron is to identify proper weights for neuronal connections which achieve the desired output in combination with inputs. This is carried out by an algorithm for back propagation [17].

B. *Back Propagation Algorithm*

Back propagation algorithm spreads output layer error into the cached layers and fixes weight changes from output layer to input layer through a network. The main purpose of the algorithm is for changing weight to minimize the output error. The algorithm of reverse propagation is based on a downward gradient. In each stage, the gradient of the goal is calculated, which direction the negative gradient shows how the surface falls more quickly and how far the direction is valid [18].

In the gradient descent direction (negative gradient), the Classic back propagation changes weights in order to reduce output faster. If function decreases, it does not necessarily lead to the fastest convergence. But search with directions that because faster convergence is done in combination with gradient algorithms. Each iteration is optimized for most conjugated algorithms [18].

V. PROPOSED MACHINE LEARNING ALGORTIHM

The Network Intrusion detection system is built by planning and implementing a hybrid machine learning model. The CICIDS2017 dataset is used to train and test the proposed model. The algorithms used to develop the Network Intrusion detection system (hybrid machine learning model) are Principal Component Analysis/PCA (unsupervised machine learning) used for dimensionality reduction, Kohonen Self Organizing Maps/SOM (unsupervised machine learning) used for visualization and clustering and Artificial Neural Networks/ANN (supervised Machine learning) used for regression and classification. The details of the proposed algorithm are given below:

- Split the CICIDS2017 data set "X" with "k" features to:
- Training data set = {"X_train" without the output label}
- Testing data set = {"X_test" and "Y_test"}
- Pre-process the training and testing data set "X_train" and "X_test".
- Select "n" features among "k" for dimensionality reduction using Principal Component Analysis (PCA) for the data set "X_train". (n<k).
- Apply Self Organizing Maps (SOM) on the extracted features to obtain a "M*M" 2-dimensional map with clusters formed.
- Visualize and Mark the clustered data to {1,0}, Let "Y_Som" be the marked output of the clustered data.
- Train the Artificial Neural Network (ANN) with "X_train" as the input data and "Y_Som" as the output label.
- Test the Artificial Neural Network (ANN) with the testing dataset "X_test" to obtain a predicted output "Y_Pred".
- Build a Confusion Matrix for obtaining the model accuracy by comparing "Y_test" and the predicted result "Y_Pred".

A. *Split the CICIDS2017 data set*

The CICIDS dataset "X" with "k" features is split into the training data set "X_train" with no output labels consisting of 11567 instances and testing data set "X_test" and "Y_test" 2830 instances.

B. *Pre-process the training and testing data set "X_train" and "X_test"*

The data set is passed through a community of pre - processing activities in order to provide more appropriate data for hybrid models. The following summarizes these transactions:

- Remove information about sockets: Since the source and destination host IP addresses and port names are contained in the original dataset, it is important to delete that information to give unbiased detection, when the

use of that information may lead to overfitting training in relation to the socket. However, it is more important to let the classifier learn from the features of the packet itself so that any host containing similar information is filtered out regardless of its socket data.

- Remove white spaces: White spaces include some of the multi-class labels in the dataset. These white spaces lead to different classes because the current value is different from other tuples in the same class.
- Label encoding: Multi-class label values of the attack string are provided in the data set. Therefore, these values should be encoded in numerical values in order to allow the classifier to learn the class number to which each multiple belongs. (BENIGN = 0, ATTACK = 1)
- Standardization of data: numerical data in the dataset varies in range, posing some challenges for the classifier to compensate for those differences during training. Therefore, it is important to standardize each attribute 's values such that, while the maximum is one, the minimum value is zero in each attribute. This provides the classifier with more homogeneous values, thus retaining the relative difference between the attributes.
- Remove / substitution of missing and infinity values: The dataset includes tuples as the missing values, and infinity values; it has been targeted in two ways that s generates two datasets.

C. Select "n" features among "k" for dimensionality reduction using Principal Component Analysis (PCA) for the data set "X_train". (n<k).

Principal Component analysis (PCA) is applied to the "X_train" for feature extraction and dimensionality reduction, with 6 features being extracted out of 73. The sklearn.decomposition.PCA library is used to implement Principal Component analysis (PCA) in this paper.

*D. Apply Self Organizing Maps (SOM) on the extracted features to obtain a "M*M" 2-dimensional map with clusters formed.*

After the feature extraction using Principal Component Analysis (PCA) on the training data set "X_train", It is passed as the input for the Self Organizing Map (SOM). The Self Organizing maps goes through 500 iterations with a learning index of 0.5 to form a 10*10, 2-Dimensional Map. The Minisom library is used to implement Self Organizing Maps in this paper.

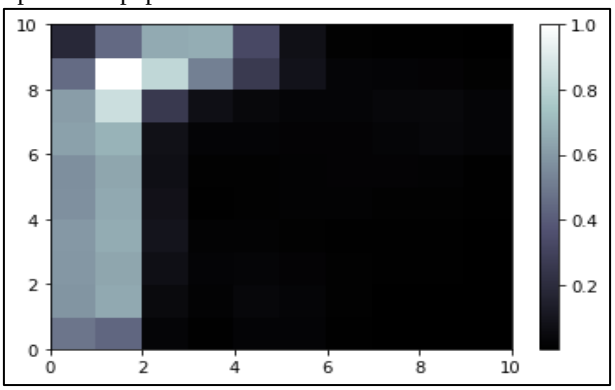

Fig. 3: 10*10 - Self Organizing Map

E. Visualize and Mark the clustered data to {1,0}, Let "Y_Som" be the marked output of the clustered data.

The produced Self Organizing Map (SOM) clusters the "X_train" data based on the mean inter neuron distances into two parts, i.e. BENIGN and ATTACK which can be visualized. By visualizing the Self-Organized Map, it is noticed all the ATTACKS are clustered and mapped to the coordinate (0,9), whereas the BENINGS are clustered and mapped to other coordinates. This clustered data is mapped to {1,0}, where 1 = ATTACK and 0 = BENIGN. Let "Y_Som" be the marked output of the clustered data.

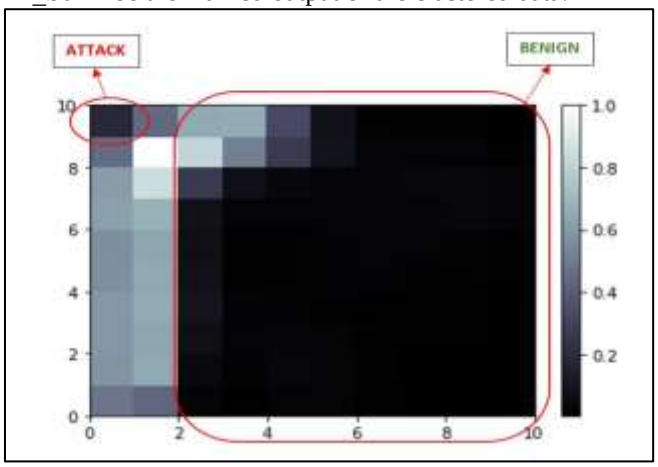

Fig. 4: 10*10 - Clusters in Self Organized Maps

F. Train the Artificial Neural Network (ANN) with "X train" as the input data and "Y_Som" as the output label

Once the output label "Y_Som" is obtained we use it along with "X_train" to train our classifier Artificial Neural Network (ANN). The keras library is used to implement the Artificial Neural Network (ANN)in this paper with 72 input dimensions, 1024 neurons in the first layer with "Relu" being its activation function, 768 neurons in the hidden layer with "Relu" being its activation function and 1 output layer with sigmoid being its activation function.

G. Test the ANN with the testing dataset "X_test" to obtain a predicted output "Y_Pred"

Once the classifier Artificial Neural Network (ANN) is trained on the data "X_train" and "Y_Som". The testing dataset "X_test" is passed as the input to the Artificial Neural Network (ANN) for classification, i.e. "BENIGN" or "ATTACK". Let "Y_Pred" be the classification result obtained from the classifier Artificial Neural Network.

H. Build a Confusion Matrix for obtaining the model accuracy by comparing "Y_test" and the predicted result "Y_Pred"

Once the predicted output "Y_Pred" is obtained, a confusion matrix is built by comparing "Y_test" and "Y_Pred" to obtain the model accuracy. The sklearn. metrics library is used to implement the confusion matrix in this paper.

VI. RESULTS AND DISCUSSIONS

The paper is planned and implemented successfully. A confusion matrix is built to evaluate the developed hybrid machine learning algorithm in this paper. A confusion

matrix allows the right and incorrect classifications of each class to be further broken down.

An accuracy of 83% on the test data was achieved by analyzing the confusion matrix, with a False Positive of 9.40% and False Negative of 6.64%.

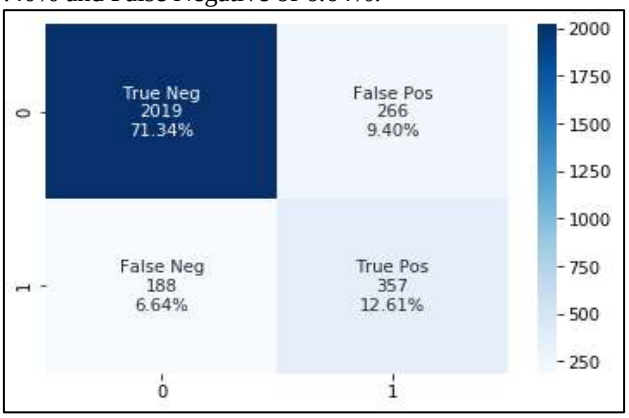

Fig. 5: Confusion Matrixs

VII. Conclusion

An intrusion detection system plays a very important role in the cyber security domain to prevent attacks on the networks. The performance depends directly on the decision-making engine. The system needs to be implemented as an anomaly detection with a learning system, in order to increase system flexibility rather than signature-based detection.

Therefore, in this paper, a novel idea was developed to build a Network Intrusion detection system by developing of a hybrid machine learning model on the CICIDS2017 dataset.

A combination of 3 Machine learning Algorithms (Supervised and Unsupervised) where used for network Intrusion Detection. An accuracy of 83% on the test data was achieved by analyzing the confusion matrix, with a False Positive of 9.40% and False Negative of 6.64%

References

[1] J. B. P. H. a. G. M. D. Batchelder, "Microsoft Security Intelligence Report - Volume 17".
[2] K. W. a. S. Stolfo, "Anomalous payload-based network intrusion detection".
[3] J. L. C. N. M. J. J. a. J. U. Z. Zhang, "HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification".
[4] Farzan, "Intrusion Detection System Using Self Organizing Map Algorithms,".
[5] M. Ringnér, What is principal component analysis .
[6] J. Shlens and M. View, "A Tutorial on Principal Component Analysis".
[7] K. T, Self-organizing maps (2nd edition).
[8] V. D. S. Krose B., "An introduction to neural networks, The University of Amsterdam".
[9] R. Lasri, "Clustering and classification using a self-organizing MAP: The main flaw and the improvement perspectives," 2016 SAI Computing Conference (SAI), 2016.
[10] E. A.A, "Shumsky S.A., Neurocomputing et son application dans l'économie et les affaires: Proc. ıłłɛɋ allocation".
[11] W. F, " Neural Computing. Theory and Practice. Théorie et pratique".
[12] S. J. R. a. P. Norvig, Artificial intelligence: A modern approach (international edition.
[13] S. Haykin, Neural networks: A comprehensive fundation.
[14] F. Haddadi, S. Khanchi, M. Shetabi and V. Derhami, "Intrusion Detection and Attack Classification Using Feed-Forward Neural Network".

