Network Traffic Vulnerability Analysis using Machine Learning- A comparative approach

Shrabani Mallick[1†] Dharmender Singh Kushwaha[2††]

Dr. B. R. Ambedkar Institute of Technology, Port Blair, India1, Motilal Nehru National Institute of Technology,
Allahabad, India

Summary

With the increase in use of web application to enable business and social networking rapid application development and deployment has been commonplace, this has increased the risk of potential network threats vulnerabilities. Thus, one the biggest challenge of the time is insecure codes running on various servers in the network making the network vulnerable and susceptible to network security breaches. Various Machine Learning using supervised and unsupervised models have been widely used to delve deep into network access log data to discover network vulnerabilities. This paper presents a comparative study of 3 machine learning approaches Naïve Bayes, k-Nearest Neighbour (kNN) and an Artificial Neural Network (ANN) to analyze the network access logs for vulnerability, particularly involving application access over network. The results are quite convincing with Naïve Bayes model with an accuracy score of 94% as compared to K- Nearest Neighbour with an accuracy of 85.2%. MLP is also reckons an accuracy score of 90.37% with very high prediction rates. The training times of MLP is of course high due to the number of epochs.

Key words:

Naïve Bayes, kNN, ANN, Supervised, Unsupervised learning

1. Introduction

The problem of insecure software is perhaps the most important technical challenge of our time. The dramatic rise of web applications enabling business, social networking etc has only compounded the requirements to establish a robust approach to writing and securing our Internet, Web Applications and Data[1]. For reasons not limited to these like insufficient security testing of web applications, race in rapid application development/ deployment to make businesses omni present, exponential increase in number of network users accessing numerous web applications, the chances of breaching network security have risen manifold. Analysis of Network Access logs and prediction of network traffic has been widely researched in recent past and has newly attracted significant number of studies.

The big world of data has been mesmerized with the buzz word of Data Science. Machine learning is the primary means by which data science manifests itself to the broader world. Machine learning is where these computational and algorithmic skills of data science meet the statistical thinking of data science, and the result is a collection of approaches to inference and data exploration that are not about effective theory so much as effective computation. Thus, using different machine learning techniques on the network access logs gives an efficient and flexible solution for network vulnerability analysis. Machine Learning approaches are categorized into supervised and unsupervised learning algorithms which have specific strengths and characteristics. Various techniques have been studied and experimented for analysing network traffic including neural networks. Similarly, various Linear and non- linear models are proposed for network traffic prediction as well. Several interesting combinations of network analysis and prediction techniques are implemented to attain efficient and effective results by various researchers which have been discussed in subsequent sections. But the efficiency of any approach depends on careful categorization of logs and then application of a suitable learning algorithm to learn susceptible patterns

This paper presents a comparative approach of three supervised and unsupervised machine learning technique – kNN, Naïve Bayes and ANN to propose an efficient technique to survey network application access logs and thereby predicting vulnerability of access traffic. technique. The reason behind selecting these techniques are they are simple, efficient yet powerful ML approaches.

The next section discusses some of the important related works carried out in the area of network traffic analysis. The subsequent two sections discusses the various web application related threats and machine learning approaches that forms the basis of the paper. The next section discusses the proposed work and results achieved followed by conclusion section.

2. Related Works

The Open Web Application Security Project (OWASP) [3] is a worldwide free and open community focused on improving the security of application software. Our mission is to make application security ―visible‖, so that people and organizations can make informed decisions about application security risks. Every one is free to participate in OWASP and all of our materials are available under a free

Manuscript received June 6, 2020
Manuscript revised June 20, 2020

and open software license. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work. Nikita Gupta et al. [1] presented rough set theory to reduce dimensionality as well as classification. Authors collected network traffic data set from NSL-KDD database. They concluded that the rough set theory approach achieved high accuracy to reduce dimensions of attribute set and also to detect intrusion.

Jashan Koshal et al. [2] proposed hybrid model for developing the intrusion detection system by combining C4.5 decision tree and Support Vector Machine (SVM) approaches. They collected data set from KDD cup. The pre-processing of data reduced the dimensionality of entire network traffic data set using feature selection method. Rohit Khandelwal et al. [4] uses a perceptron to analyse DOS and DDOC attacks.

Mu et al. [5] uses statistical features to identify the network traffic efficiently without detecting the payload of every packet. In their work in order to filter the more effective statistical features to construct the neural network, the efficiency of each statistical feature has been analysed and results are presented to identify best NN models. Cheng et al.

[6] propose an automatic signature extraction mechanism using Principal Component Analysis (PCA) technology, which is able to extract the signature automatically. In their proposed method, the signatures are expressed in the form of serial consistent sequences constructed by principal components instead of normally separated substrings in the original data extracted from the traditional methods. Yao et al.

[7] propose a new algorithm based on instantaneous parameters (instantaneous frequency and Instantaneous amplitude) analysis. The characteristic of traffic anomaly would be revealed more evidently through analysing the instantaneous parameters of the original network flow data.

Ji et al. [8] experiment to find the optimal combination between Naive Bayes and HNB, a novel model Packaged Hidden Naive Bayes (PHNB), which the number of attributes in the hidden parent is controlled through packaging idea and show that compared to HNB, PHNB significantly reduces the test time on many high-dimensional datasets, and has higher accuracy on some particular datasets. Madalgi and Kumar [9] applied

machine learning techniques to detect the different levels of congestion in as low, medium or high. The work proposes that classification by regression is more efficient than MLP in detecting the congestion for the generated data set of WS'N simulation using NS2.

3. Web Application Testing and Security Threats

There are many techniques and approaches that can be used for testing the security of web applications. Experiments and experience have shown that there are as such no right or wrong technique as to answer the question of exactly what techniques should be used to build a testing framework for making a web application fool proof against potential security vulnerabilities.

Fig.1, presents an ideal testing framework workflow [3].

Some of the common vulnerabilities that may lead to security threats as follows:

- Unsafe passwords that allows dictionary guesses
- Web spiders/robots/crawlers can intentionally ignore the Disallow directives specified in a robots.txt file [4], such as those from Social Networks [2] to ensure that shared linked are still valid. Hence, robots.txt should not be considered as a mechanism to enforce restrictions on how web content is accessed, stored, or republished by third parties.
- Account Enumeration and Guessable User Account
- SQL Injection
- Multiple gates for entry
- Credentials Transported over an UnEncrypted Channel or loosely encrypted channel (in efficient algorithms.
- Receiving and Sending data through HTTP Get/ Post methods which tells that data is transmitted without encryption and a malicious user could intercept the username and password by simply sniffing the network with a tool like Wireshark.

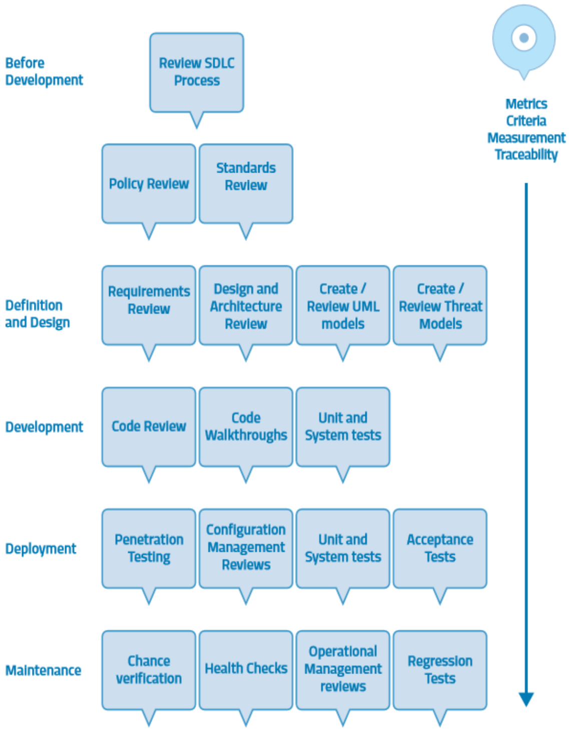


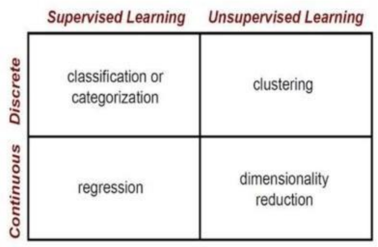
Fig. 1 OWASP Testing framework workflow [3]

4. Machine Learning Approaches

At the most fundamental level, machine learning can be categorized into two main types:

- Supervised learning and
- unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. It includes such algorithms as linear and logistic regression, multi-class classification, and support vector machines.



	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Fig. 2 The 4 quadrants of Machine Learning Algorithms

On the other hand, unsupervised machine learning is more closely aligned with true artificial intelligence — the idea that a computer can learn to identify complex processes and patterns without a human to provide guidance along the way. These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. In addition, there are so-called semi-supervised learning methods, which falls somewhere between supervised learning and unsupervised learning. Semi-supervised learning methods are often useful when only incomplete labels are available. Fig. 2, summaries the four quadrants of Machine Learning Techniques.

The proposed work of data analytics for analysing network access logs has been carried out applying the strength of machine learning (ML) techniques to infer if the intended access to the application in the network is malicious or not. In our work we have categorized the traffic into two classes – Vulnerable and Non-Vulnerable. We have used 3 ML techniques, namely

- k-Nearest Neighbour technique – One of the most known classification and regression algorithm used in ML. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). The data for KNN algorithm consists of several multivariate attributes name that will be used to classify
- Naïve Bayes technique - A Naive Bayes classifier is a probabilistic machine learning model that‘s used for classification task based on Bayes‘ theorem. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.
- Multi-Layer Perceptron (ANN) - A multilayer perceptron is a class of feedforward artificial neural network. An MLP consists of, at least, three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.

5. Proposed Work

The proposed work is divided into three broad phases – ExperimentalSetup, Pre-processing and Analysis. Fig3, shows the different steps involved in our approach. The different steps are described as follows:

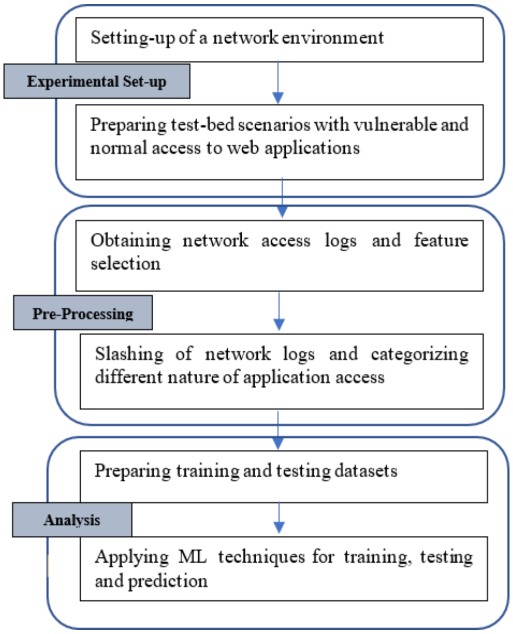

Setting-up of a network environment
Experimental Set-up
Preparing test-bed scenarios with vulnerable and normal access to web applications
Obtaining network access logs and feature selection
Pre-Processing
Slashing of network logs and categorizing different nature of application access
Preparing training and testing datasets
Analysis
Applying ML techniques for training, testing and prediction

Fig. 3 :Process flow of proposed approach

5.1 Network and Testbed Setup

The network environment has been set using the following configuration. Fig. 4, shows the Network Environment used

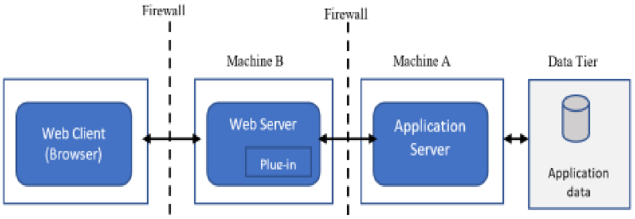


Fig. 4 Experimental Network Set-up

The test bed was generated for a variety of accesses to the web applications.
Dataset: • HTTP traffic from a university network, 24h, 200 clients, 10 domains, 800+ requests. Fig. 5 shows the graphical view of number of attacks under different categories.

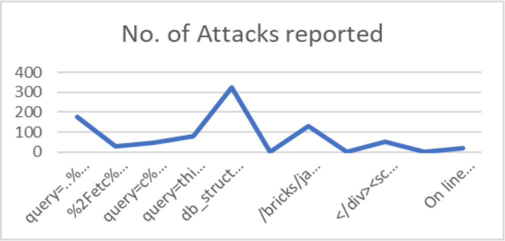


Fig. 5 Number of attacks under different categories

Fig. 6 shows the percentage of attacks at different risk levels for the generated data sets

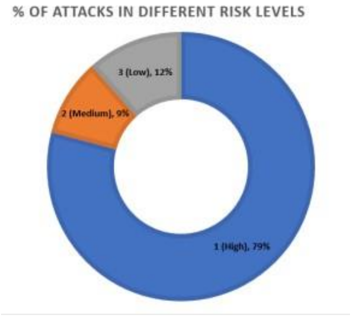


Fig. 6 % of Attacks in different Risk Levels

5.2 Preprocessing of Network Access Logs and Feature Selection

The preprocessing of logs was done using Packetbeat. Packetbeat takes care of a variety of incantations to get your data into proper shape for search and analysis in real time, on target servers. Fig. 7 shows the different steps in pre-processing of network logs using Packetbeat.

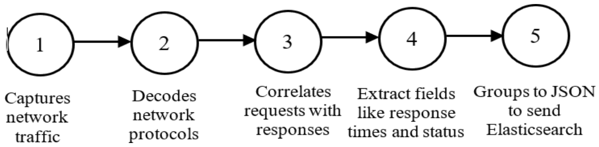

1
Captures network traffic
2
Decodes network protocols
3
Correlates requests with responses
4
Extract fields like response times and status
5
Groups to JSON to send Elasticsearch

Fig. 7 Pre-processing of network logs using Packetbeat

Apart from the fields extracted by Packetbeat, the pre-processed logs are subjected to a keyword extractor to slash the logs to understand various keywords matching potential threat keywords as shown in Table 1. The Risk level are to be read as 1 (High), 2 (Medium) and 3(Low).

Table 1: Categories Of Attacks

Attack	Description	Risk Level	No. of Attacks reported
query=..%2F..%2FWEB-INF%2Fweb.xml HTTP/1.1	Attempt to access Web-INF file	1	178
%2Fetc%2Fpasswd HTTP/1.1" 200 5826	Unauthorised password fetch	2	29
query=c%3A%2FWindows%2Fsystem.ini HTTP/1.1" 200 2945	Attempt to access system.ini file	1	45
query=thishouldnotexistandhopefullyitwillnot HTTP/1.1"	Anonymous queries- dictionary guessings	3	78
db_structure.php?db=inject&ajax_request=true	SQL Injections	1	323
/bricks/javascripts/jquery.js	Running unauthorised java scripts	1	132
</div><script>alert(1);</script><div>	Cross site scripting	2	51
On line <b> method POST url: http://192.168.12.12:81/bricks/content- 5/index.php	Parameter manipulation or lack of exception handling and potential areas for further exploit	3	21

5.3 Analysis of Network Access Logs

The analysis phase attempts to classify the incoming traffic into two classes – Vulnerable‘ and _Non-Vulnerable‘. There are a number of possible models for such a classification task, but here we have used the three models-

- k-NearestNeighbour
- NaïveBayes
- ANN

The next section presents the results and discussion of our approach

6. Results and Discussion

As discussed in the previous section the analysis of the network logs is carried out using three classification models – kNN, Naïve Bayes and ANN. The classes are be separated by a straight line through the plane between them, such that points on each side of the line fall in the same group. The optimal values for these model parameters are learned from the data which are used in training the model. The features used as input parameters are as under-

Features: {send pkts, receive pkts, access type, protocol, total_bandwidth_consumed, keywords, risk_level} Labels: { Vulnerable‘ , Non-Vulnerable}

Anaconda Python‘s scikit learn is used as the data analytic tool for the analysis purpose.
import cross_validation, neighbors
from sklearn.cross_validation
import train_test_split x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)

The dataset has been split using 80:20 principle for training and test data. For the kNN classification, the k value has been taken as 1. Fig. 8 shows error curve for the k-value. As observed in the graph, the error rate is almost zero when k=1, this enables overfitting of the classification boundary curve to train the model better during the training phase. The model was trained using 80% of the data and tested for 20% of the data. The classification plot is shown in Fig.9.

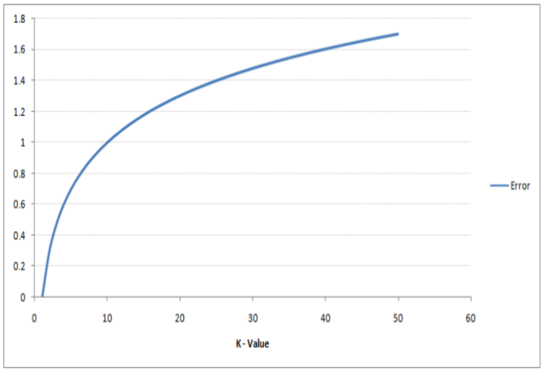


Fig. 8 Error Curve for different values of K

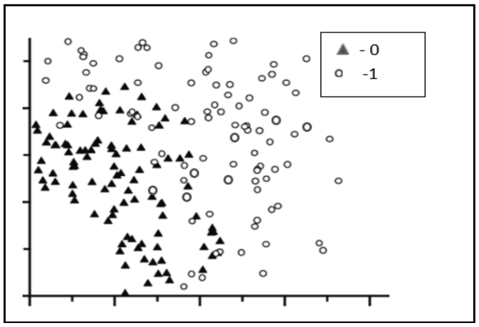


Fig. 9 Classification using k-NN (0 -Vulnerable and 1- Non-Vulnerable)

The second model used for the analysis is the Bayes classification model. Since there are 2 classes ω1 and ω2 the data should belong to one of those classes, based on training data whose class membership information is already known. Using Bayes rule, the probability that a new data x belongs to class ωi is given by-

$P(\omega i|x)=\rho(x|\omega i)P(\omega i)\rho(x)=\rho(x|\omega i)P(\omega i)\sum cl=1\rho(x|\omega l)P(\omega l)$

where ρ is a density function for continuous values. That is, $\rho(x|\omega i)$ is a class-conditional density, $P(\omega i)$ is a prior probability, $\rho(x)$ is an evidence which can be usually ignored, and $P(\omega i|x)$ is a posterior probability. We now compare the probability values of $P(\omega i|x)$ for each class and make a decision by taking one with higher probability as following

$P(\omega i|x)\geq P(\omega j|x)\forall j=1,\ldots,c$
$\Leftrightarrow\rho(x|\omega i)P(\omega i)\rho(x)\geq\rho(x|\omega j)P(\omega j)\rho(x)\forall j=1,\ldots,c$
$\Leftrightarrow\rho(x|\omega i)P(\omega i)\geq\rho(x|\omega j)P(\omega j)\forall j=1,\ldots,c$

In our case we have only two class classification problem. Let g1(x)=ρ(x|ω1)P(ω1) and g2(x)=ρ(x|ω2)P(ω2). Fig 10 shows the Bayes classification is shown in 3-dimensional feature space.

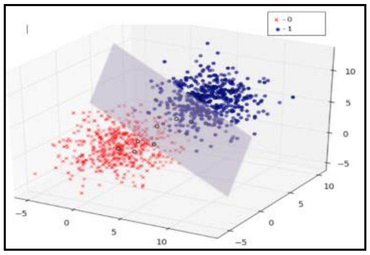


Fig. 10 Classification using Bayes Classification Technique (0 - Vulnerable and 1- Non-Vulnerable)

For the MLP we have used 01 input layer, 02 hidden layers, and 01 output layer. We have used hyperbolic tangent activation function. Fig. 11 shows the classification result of the MLP. The key features of an activation function are to be bounded, parameterized, and, usually, nonlinear.

We have used a multilayer feedforward, with an hyperbolic tangent activation function so that it is monotonically increasing and just for the output layer – a linear mapping. The hyperbolic tangent function is antisymmetric sigmoid function. It has been observed that the Back propagation algorithm, learns faster when the activation function is an antisymmetric sigmoid than when it is a non symmetric one.

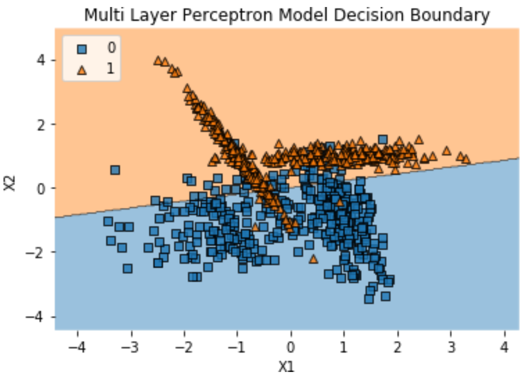


Fig. 11 Classification result of the MLP (0 -Vulnerable and 1- Non-Vulnerable)

After the data is subjected to the three models, the classification results obtained are shown in Table II. The training and testing times of the three approaches is shown in Table III.

Table 2: Classification Results

MODEL	ACCURACY SCORE
k-Nearest Neighbour	85.29411764705882
Naïve Bayes	94.11764705882353
ANN	90.375

Table 3: Training Testingtime

MODEL	TRAINING TIME (MS)	TESTING TIME (IN MS)
k-Nearest Neighbour	0.05	0.005
Naïve Bayes	0.032	0.01
ANN	0.42	0.08

The Box plot at Fig.12 shows the bandwidth consumed by the traffic data with respect to mean of the data. The descriptive statistical analysis of the total bandwidth (BW) consumed data shown in Table IV reveals that 75% of the population is above the mean bandwidth consumed which could be vulnerable for DDOS attack. This is evident from the MLP plot as the concentration of blue squares is more towards the decision boundary of vulnerable class, thereby showing that feature selection holds good.

Table 4: Statistical Desc Of Total Bwconsumed By The Network Traffic

MODEL	STATISTICS	DESCRIPTION
Total data	868	
Mean	869.77	
Std	3089.95	
25% of data	5.75	

50 % of data	62.0	
75% of data	461.0	* Could be vulnerable for DDOS attack

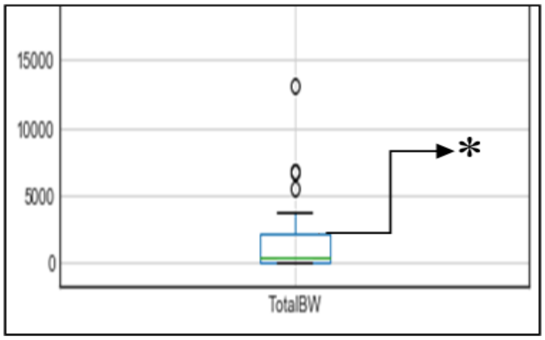


Fig. 12 Box Plot of Total Bandwidth consumed

7. Conclusion

Insecure applications used over network may result in security breach of any organization. Lack of proper testing at all levels of SDLC lay leave vulnerability holes in the web application which could be disastrous for businesses having network presence. Thus, the proposed conducts a vulnerability analysis webapplicationaccesslogsusingMLtechniques.MLinvolves building mathematical models to help understand data. These models can adapt to observed data. Once these models have beenfittopreviouslyseendata,theycanbeusedtopredictand understand aspects of newly observed data. Understanding the problem setting in machine learning is essential to using these tools effectively. The proposed work carries out the analysis through 3 models. The results are quite convincing with Naïve Bayesmodelwithanaccuracyscoreof94%ascomparedtoK-Nearest Neighbour with an accuracy of 85.2%. MLP is also reckons an accuracy score of 90.37% with very highprediction rates. The training times of MLP is of course high due to the number ofepochs.

References

[1] (N.Gupta, N.Singh, V. Sharma, T. Sharama, A.S. Bhandra, Feature Selection and Classification of intrusion detection using rough set (International Journal of Communication Network Security, 2013)ISSN: 2231–1882,Volume-2,Issue-2G.Eason,B.Noble,andI.N.Sneddon,

[2] ―On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,‖ Phil. Trans. Roy. Soc. London, vol. A247, pp. 529– 551, April 1955.

[3] J. Koshal, M. Bag, Cascading of C4.5 Decision Tree and Support Vector Machine for Rule Based Intrusion Detection System (Computer Network and Information Security, 2012) , 8,8-20

[4] Open Web Application Security Project, Testing Guide Release 4.0 Project Leaders: Matteo Meucci and Andrew Muller Creative Commons (CC) Attribution Share-Alike Free version at http://www.owasp.org.

[5] Rohit Khandelwal, Abhinav Srivastava, Md. Ejaz Uddin, Network Traffic Classification using Artificial Neural Network, Journal of Harmonized Research (JOHR) 2(1), 2014, page15-19

[6] Cheng Mu, Changzhi Zhang, Xiaohong Huang, Yan Ma, The efficiency analysis of the statistical feature in network traffic identification based on BP neural network, 5th IEEE International Conference on Broadband Network & Multimedia Technology, 17-19 Nov.2013

[7] Mu Cheng ; Huang Xiaohong ; Wu Jun ; Ma Yan, Network traffic signature generation mechanism using principal component analysis, IEEE China Communications (Volume: 10 , Issue: 11 , Nov. 2013)

[8] Xingmiao Yao ; Peng Zhang ; Jie Gao ; Guangmin Hu, Detection of Network Traffic Anomaly Based on Instantaneous Parameters Analysis, IEEE International Conference on Communication Technology, 27-30 Nov. 2006

[9] Yaguang Ji ; Songnian Yu ; Yafeng Zhang, A novel Naive Bayes model: Packaged Hidden Naive Bayes, 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, 20-22 Aug.2011

[10] Jayashri B. Madalgi ; S. Anupama Kumar, Congestion detection in wireless sensor networks using MLP and classification by regression, 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 21-23 Dec.2017

[11] https://en.wikipedia.org/wiki/Multilayer_perceptron

[12] F. Marini, Chapter 3 Neural Networks, Comprehensive Chemometrics Chemical and Biochemical Data Analysis2009, Pages 477-505

[13] Ehsan Fathi, Babak Maleki Shoja in Handbook of Statistics, Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications, 2018

[14] Viet-Thanh Pham, Tomasz Kapitaniak, Chaos in a System With Parabolic Equilibrium in Recent Advances in Chaotic Systems and Synchronization, 2019

Dr. Shrabani Mallick, is Lecturer in the Dept of Computer Science Engineering at Dr. B. R Ambedkar Institute of Technology, Port Blair Andaman and Nicobar Islands. She received her Ph.D degree from Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh India and has 18 years of teaching experience. Her research interest is Distributed Computing, Machine Learning, Internet of Things and Software Engineering.

Dr. Dharmender Singh Kushwaha is Professor in the Dept of Computer Science Engineering at Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh India. He has 28 years of teaching experience and is a life member of IEEE and Computer Society of India. His research interest is Distributed System and cloud Computing, Web Services and Data Structures, Internet of Things and Software Engineering.