Improving Intrusion Detection System Using an Extreme Learning Machine Algorithm

M.S. Abirami, Shivam Pandita, Tanvi Rustagi

Abstract*: An Intrusion Detection System (IDS) is a system, that checks the network or data for abnormal actions and when such activity is discovered it issues an alert. Numerous IDS techniques are in use these days but one major problem with all of them is their performance. Various works have been done on this issue using support vector machine and multilayer perceptron. Supervised learning models such as support vector machines with related learning algorithms are used to analyze the data which is used for regression analysis and also classification. The IDS is used in analyzing big data as there is huge traffic which has to be analyzed to check for suspicious activities, and also be successful in doing so. Hence, an efficient and fast classification algorithm is required. Machine learning techniques such as neural networks and extreme machine learning are used. Both of these techniques are highly regarded and are considered one of the best techniques. Extreme learning machines are feed forward neural networks which have one hidden layer and no back propagation used for classification. Once the intrusion is detected using IDS through ELM then we are also going to detect the type of intrusion using the Random Forest Technique (Multi class classification) efficiently with a higher rate of accuracy and precision. The NSL_KDD dataset which is very well-known used for the training as well as testing of these IDS algorithms. This work determines that compared to artificial neural network and logistic regression extreme learning machines provide a much better rate of intrusion detection, which is 93.96% and is also proven to be more efficient in terms of execution time of 38 seconds.

Index Terms*: Artificial Neural Network, Extreme Learning Machine, Logistic Regression, False Alarms, Intrusion Detection System

I. INTRODUCTION

There is an ever growing demand of a vigorous security to be implied upon the developing technology. Despite the increase in demand of network security, the current existing solutions are still inadequate in fully securing the computer networks and internet applications against the ever-advancing threats from hackers in the form of cyber-attacks such as DOS attacks and many more [1]. Creating more advanced and adaptive IDS which are very fast and efficient is more important now than ever before. The old security techniques like user authentication, data encryption and firewall are not sufficient anymore in front of the advanced intrusion attacks faced these days. Hence, a strong security defense line is the need of the hour, such as Intrusion Detection System (IDS).

Revised Manuscript Received on July 05, 2019.

M.S. Abirami, Department of Software Engineering, SRMIST, Chennai, India.

Shivam Pandita, Department of Software Engineering, SRMIST, Chennai, India.

Tanvi Rustagi, Department of Software Engineering, SRMIST, Chennai, India.

An Intrusion Detection System (IDS) is a software application that screens a network or systems for malevolent activity or policy destructions. The IDS types range in scope from large grids to even single computers. NIDS and HIDS (network intrusion detection systems and host based intrusion detection systems) are the most common groupings in IDS [9]. HIDS is an example of a system that aims to screen important OS files whereas NIDS is a system that keeps an eye on the incoming traffic of a system [2]. Intrusion is very important problem in security and it is one of the main issues of security breach. Even a tiny breach can lead to loss of large amount of data from network systems and computers in matter of seconds. Even system hardware can be damaged as a cause of intrusion. Furthermore, intrusion can also lead to huge capital losses and thereby can also be used as a major weapon in cyber war. Hence we can say that intrusion detection is very crucial and avoiding it is mandatory. We have a lot of various options for choosing intrusion detection systems but the main problem lies in how accurate they are; which depends on correct detection and also the rate of false alarm generated. This problem on accuracy needs to be checked so that false alarms can be reduced and the detection rate can be increased. This idea was the drive for this research. Thus, Logistic Regression (LR), Artificial Neural Network (ANN), and Extreme Learning Machine (ELM) are applied in this work; these methods have been used to show comparison among the various methods of classification. All the various IDS are validated on the same dataset known as the KDD [6]. This paper used the 'NSL knowledge discovery and data mining (KDD) dataset', which can be called a more efficient version of the KDD dataset and clearly sets a benchmark for the other various datasets. The background and related work are discussed in section II. In the proposed model of intrusion detection system, deep learning and machine learning techniques are applied which are explained in section III. The results are discussed in section IV. This paper is concluded in section V with references.

II. RELATED WORK

It is important for organizations to secure computer information and network details, because compromised information can cause a lot of damage. This is the reason why grave importance is given to intrusion detection system. Recently a lot of algorithms are proposed to be applied on the KDD99 dataset. This dataset could altogether lead to accuracies as high as 98.3%, but the dataset contains various limitations such as minimum test data.

The NSL-KDD dataset is the better version of the KDD cup99 data set [5]. There have been a lot of researches under various domains approved by different researchers on the

NSL-KDD dataset employing a lot of tools and practices with the universal goal of improving the intrusion detection system. There is a detailed analysis which was done on the NSLKDD dataset using the WEKA tool which uses many machine learning techniques. There was a comparison made between the NSLKDD dataset and its previous version the KDDCUP99 dataset using the 'Self Organization Map (SOM) Artificial Neural Network', there was an analysis which was conducted for the comparison of KDD99, Gure-KDD and NSLKDD.

This analysis used a lot of deep learning and data mining techniques such as Support Vector Machine (SVM), K-nearest neighbor, Decision Tree, K-Means and other algorithms. Mohammed A. Ambusaidi et. al. [6] presented the concept of Building an Intrusion Detection System using Filter-based Feature Selection algorithm. They presented mutual information based feature selection algorithm. They achieved 92% accuracy in LSSVM based algorithm trained and tested on KDD Cup-99 dataset. Yu Su et. al. [3] proposed Learning Automata based feature selection for NIDS. Learning Automaton (LA) method is a decision maker with adaptive nature. It was designed to learn the behavior of biological tissues. It constantly interacts with external environment in order to learn stochastic behavior and maximize benefits. They have used learning automata based algorithms and achieved accuracy 80% to 90% for each.

Experiment method on KDD data set was giving 94.32 percent accuracy with false positive rate of 0.7.

So many IDS has been proposed using ensemble learning. Akhilesh Kumar et. al. [11] developed an ensemble model which is based on the two classifiers Artificial Neural Network (ANN) and Bayesian Net, and they combined it using Gain Ratio with Feature Selection Technique.

III. PROPOSED METHODOLOGY

The main phases of our system which each have its own importance are shown below. Namely, the phases are – dataset, pre-processing, classification, and final evaluation.

A. NSL-KDD Dataset

The NSL-KDD data is used here which can be called a more efficient version of the KDD99 dataset [7]. There have been a lot of researches under various domains approved by different researchers on the NSL-KDD dataset employing a lot of tools and practices with the universal goal of improving the intrusion detection system. There is a detailed analysis which was done on the NSLKDD dataset using the WEKA tool which uses many machine learning techniques [10].

KDD99 has a lot of limitations which has been revealed by a lot of studies and hampered a lot of intrusion detection systems. The NSL-KDD data is used here which can be called a more efficient version of the KDD99 dataset. There are files about important and complete entries of this dataset available on the internet for the users to use.

- The records which are repetitive are removed. This will lead the classifiers to give impartial results.
- This data set contains various test and training data which enables us to perform experiments with more accuracy.
- The percentage of records in the original KDD data set is inversely proportional to amount of selected records from each difficult level group. In each record there are forty one attributes. These attributes describe the various features of the flow and a tag given to all either as 'intrusion' or as "normal".

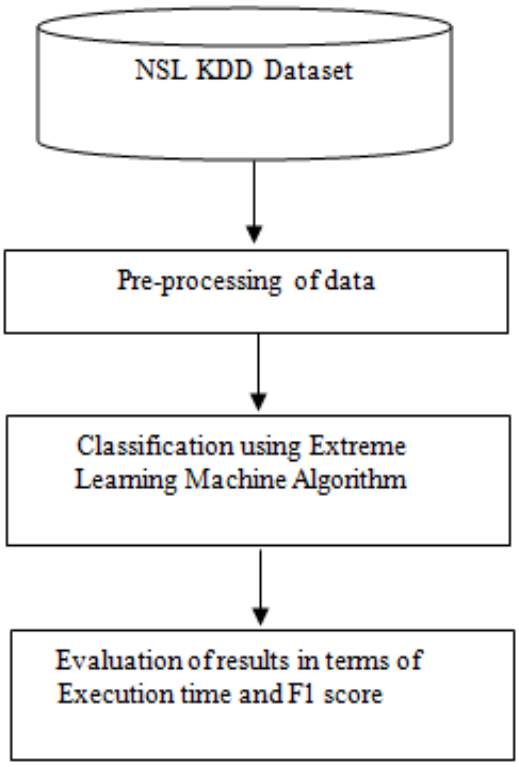

Fig. 1: Proposed Methodology for IDS

B. Preprocessing

- The original data which was available consists of 42 features. Few features which were dependant on some other feature were rejected, to avoid over fitting the model.
- Further, the dataset consisted of readings in the form-'normal' and 'anomaly' which were encoded into 0,1
- To reduce variance and standardize the data scikit pre-processing library was used to normalize the data.
- Further the data was split into a 0.2 test and remaining training set.

C. Classification

1) Classification using Logistic Regression

Logistic Regression is a statistical method for analyzing a dataset with one or more independent variables that determines a result [5]. The output of logistic regression can be measured using execution time and F1 score.

Table I: NSL-KDD Dataset

attribute	name	type
attribute 1	'duration'	real
attribute 2	'protocol_type'	type
attribute 3	'service'	type
attribute 4	'flag'	type
attribute 5	'src_bytes'	real
attribute 6	'dst_bytes'	real
attribute 7	'land'	0,1
attribute 8	'wrong_fragment'	real
attribute 9	'urgent'	real
attribute 10	'hot'	real
attribute 11	'num_failed_logins'	real
attribute 12	'logged_in'	0,1
attribute 13	'num_compromised'	real
attribute 14	'root_shell'	real
attribute 15	'su_attempted'	real
attribute 16	'num_root'	real
attribute 17	'num_file_creations'	real
attribute 18	'num_shells'	real
attribute 19	'num_access_files'	real
attribute 20	'num_outbound_cmds'	real
attribute 21	'is_host_login'	0,1
attribute 22	'is_guest_login'	0,1
attribute 23	'count'	real
attribute 24	'srv_count'	real
attribute 25	'serror_rate'	real
attribute 26	'srv_serror_rate'	real
attribute 27	'rerror_rate'	real
attribute 28	'srv_rerror_rate'	real
attribute 29	'same_srv_rate'	real
attribute 30	'diff_srv_rate'	real
attribute 31	'srv_diff_host_rate'	real
attribute 32	'dst_host_count'	real
attribute 33	'dst_host_srv_count'	real
attribute 34	'dst_host_same_srv_rate'	real
attribute 35	'dst_host_diff_srv_rate'	real
attribute 36	'dst_host_same_src_port_rate'	real
attribute 37	'dst_host_srv_diff_host_rate'	real
attribute 38	'dst_host_serror_rate'	real
attribute 39	'dst_host_srv_serror_rate'	real
attribute 40	'dst_host_rerror_rate'	real
attribute 41	'dst_host_srv_rerror_rate'	real
attribute 42	'class'	0,1

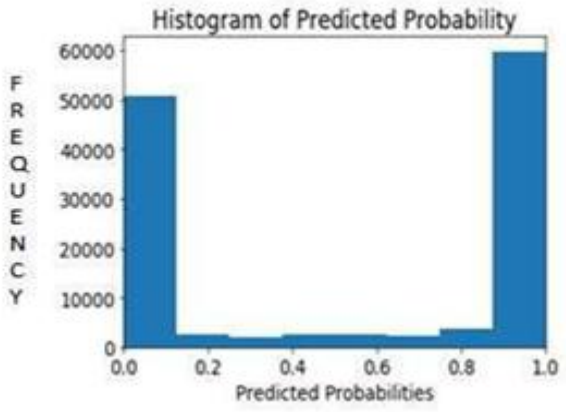

Fig. 2: Histogram of Predicted Probabilities

Table II: Result of Logistic Regression

Resultant F1 score	0.9498782018159002
Resultant confusion matrix	TP=54807, FP=3084 TN=2986 , FN=9014
Required time of execution	28.48 econds

2) Classification using Artificial Neural Network

An A*rtificial Neural Network* is basically a network of simple nodes called artificial neurons, which receive input, modify their internal state (*activation*) in terms of weights according to that input, and produce output conditional on the input and activation function applied.

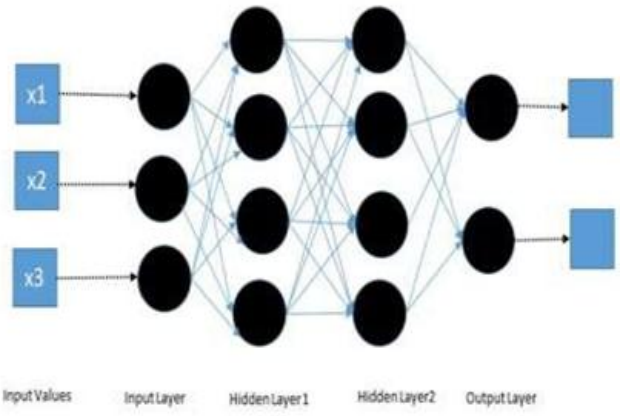

Fig. 3: Artificial Neural Network

- The presented model was trained on our chosen dataset. The model contains two hidden layers to perform the network analysis.
- The activation function for training the hidden layers was the rectifier function and to train the output layer the sigmoid activation function was used.

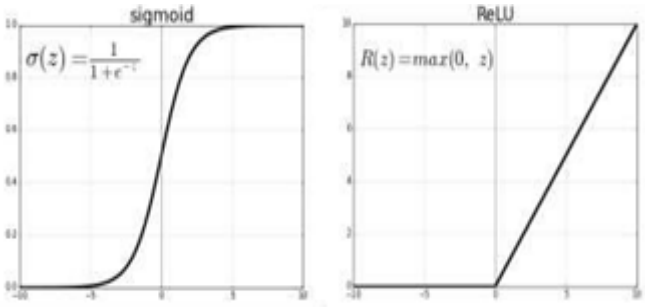

Fig. 4: Graph of Sigmoid and ReLU

In Figure 5, the graph depicts how the accuracy of the model increases proportionally with the number of epochs.

In Figure 5, the graph depicts how the accuracy of the model increases proportionally with the number of epochs.

Table III: Result of ANN

Resultant F1 score	0.7801627142115284
Resultant confusion matrix	TP=8449, FP=4383 TN=697 , FN=9014
Required time of execution	13 seconds per epoch

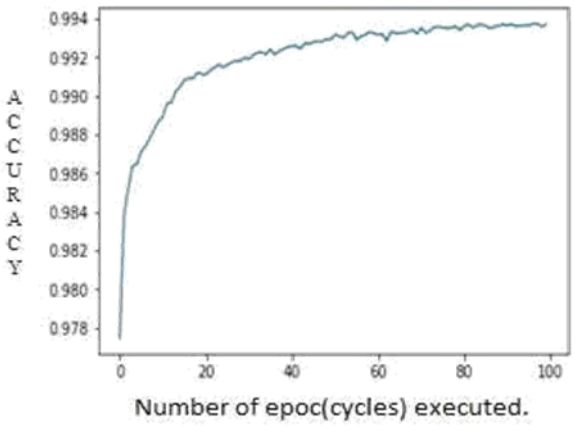

Figure 5: Accuracy vs. Number of Epochs

3) Classification using Extreme Learning Machine

The Extreme Learning Machine (ELM) is a specific kind of machine learning system in which a single layer or multiple layers apply. The ELM includes numbers of hidden neurons where the input weights are allotted randomly. Extreme learning machines use the random projection and early perceptron models to do detail problem-solving.

In theory, the Extreme Learning Machine algorithm (ELM) has extremely fast learning speed and also provides great performance results. Unlike most conventional NN learning algorithms, the ELM does not use a gradient-based technique [14]. In this method, all the parameters are tuned once. This algorithm does not need iterative training.

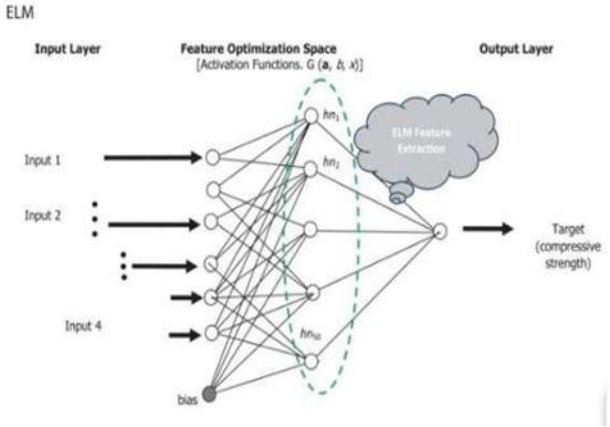

Fig. 6: Extreme Learning Machine

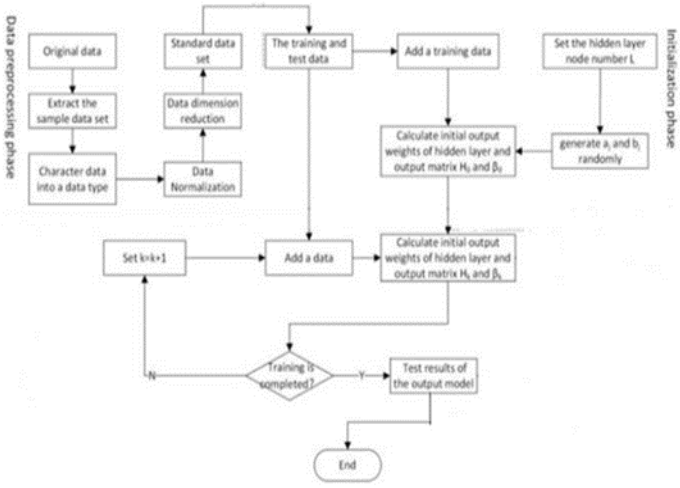

Fig. 7: Architecture of Extreme Learning Machine

ELM algorithm is one of the simple and best implementation algorithms. Also the ELM algorithm has great results with minimum computational time [15].

4) ELM implementation steps

- Generate the weights matrix W for the input layer using random numbers
- Calculate the output matrix and need to activate the output matrix. Then any desired activation function has to be chosen

$$H = W * X \qquad (1)$$

- Calculate the Moore-Penrose pseudo-inverse
- Calculate the output weight matrix beta

$$G^{+} = (G^{T} * G)^{-1} * G^{T} \qquad (2)$$

- Then the output matrix calculation is repeated for the testing dataset, creating a new H matrix. After that, the result matrix O is created by using the already known beta matrix

$$\beta = H^{+} * T \qquad (3)$$

$$O = H * \beta \qquad (4)$$

- The Soft Max algorithm is used to transform O matrix. Then matrix O is compared with matrix T using the Winner Takes All algorithm

After applying all these steps to the pre-processed dataset, it is found that the

ELM algorithm is showing the result as shown in table IV.

Table IV: Result of ELM

Resultant F1 score	0.9396825396825397
Required time of execution	38 seconds

IV. RESULTS

Results can be visualized clearly based on two major factors which are the time taken by an algorithm to execute and the accuracy of predicting the intrusion.

The graph as shown in Figure 8 depicts the time taken by all the three algorithms on the NSL-KDD dataset. The graph clearly shows that ANN takes more time compared to ELM and logistic regression.

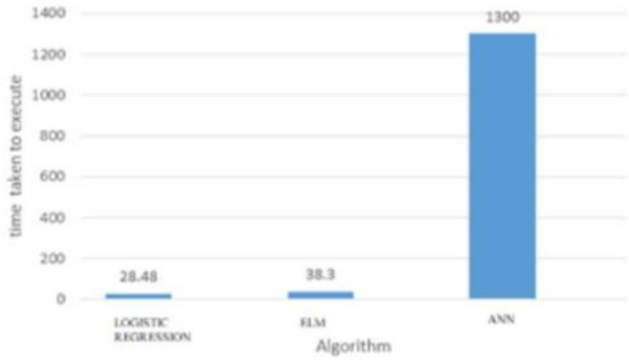

Figure VIII: Comparison of Execution time of LR, ELM, and ANN

The graph as shown in Figure 9 depicts the F1 score of all the three algorithms on the NSL-KDD dataset. It clearly shows that ELM produces best F1 score of 0.94 as compared to LR and ANN.

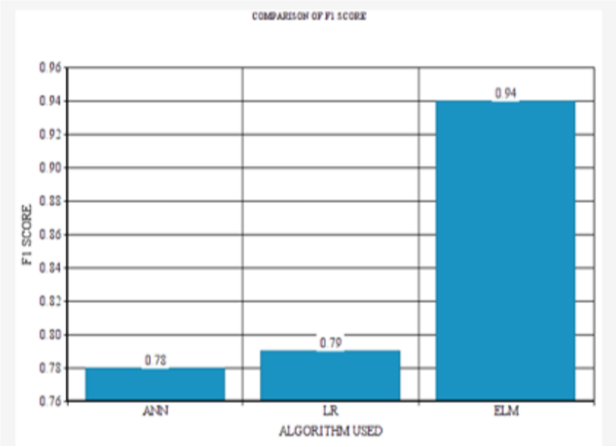

Fig. 9: Comparison of F1 score of LR, ELM, and ANN

V. CONCLUSION

Logistic Regression is applied on the NSL-KDD data set and results an accuracy of 94% and it took 28.48 seconds time to complete its execution. As there is large number of features, this model is not suited. Then the Artificial Neural Network algorithm is applied on the NSL-KDD dataset and results an accuracy of 98% and it took 1300 seconds.

As ANN takes much time to execute and due to large data size, it is also not that much suited for the intrusion detection system.

Extreme Learning Machine algorithm is one of the most efficient machine learning algorithms in the field of neural networking. It works well on very large datasets. Because of the non-iterative training, initially all the parameters are tuned. This results the speed of training dataset. Its implementation is simple to understand, and it can be used to solve complex problems.

Finally, ELM results 0.93 as F1 score and took 38 seconds to execute. So, this is the most suitable and favorable algorithm for Intrusion Detection System.

The future works of this paper include multiclass classification using random forest. The various types of intrusion will be predicted along with the intrusion alert. This work will make current system as a multitasking Intrusion Detection System.

REFERENCES

1. Iftikhar Ahmad, Mohammad Basheri, Muhammad Javed Iqbal, Aneel Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection", IEEE Access, 6, 33789-33795, 2018.
2. Kinam Park, Youngrok Song, Yun-Gyung Cheong, "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm", in IEEE Fourth International Conference on Big Data Computing Service and Applications, 282-286, 2018.
3. Yu Su ; Kaiyue Qi ; Chong Di ; Yinghua Ma ; Shenghong Li, "Learning Automata Based Feature Selection for Network Traffic Intrusion Detectio", in IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018.
4. El Mostapha Chakir, Mohamed Moughit , Youness Idrissi Khamlichi, "A Real-time Risk Assessment Model for Intrusion Detection Systems", in International Symposium on Networks Computers and Communications (ISNCC), 2017.
5. Yan Zhang, Chong Di, Zhuoran Han, Yichen Li, Shenghong Li, "An Adaptive Honey-pot Deployment Algorithm Based on Learning Automata ", in IEEE Second International Conference on Data Science in Cyberspace (DSC), 2017.
6. Mohammed A. Ambusaidi, Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan, "Building an Intrusion Detection System using a Filter-based Feature Selection Algorithm", IEEE Transactions on Computers, 65(10), 2986-2998, 2016.
7. L.Dhanabal, S.P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446-452, 2015.
8. Iftikhar Ahmad ; Fazal e Amin, "Towards Feature Subset Selection in Intrusion Detection", in IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, 68-73, 2015.
9. G. Malik, M. Tarique, "On Machine Learning Techniques for Multi-class Classification", International Journal of Advancements in Research and Technology, 3(2), 6-9, 2014.
10. Qais Saif Qassim, Ahmed Patel, Abdullah Mohd Zin , "Strategy to Reduce False Alarms in Intrusion Detection and Prevention Systems", International Arab Journal of Information Technology, 11(5), 2014.
11. Akhilesh Kumar, Shrivas Bilaspur, Amit Kumar Dewangan, "An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set ", International Journal of Computer Applications, 99(15), 2014.
12. Kumpati S. Narendra , Mandayam A.L. Thathachar, "Learning Automata: An Introduction (Dover Books on Electrical Engineering)", 2012.
13. S.J. Horng, M.Y. Su, Y.H. Chen, T.W. Kao, R.J. Chen, J.L. Lai, and C. D. Perkasa, "A Novel Intrusion Detection System based on Hierarchical Clustering and Support Vector Machines," Expert Systems

with Applications, vol. 38, no. 1, pp. 306–313, 2011.

14. Guang-Bin Huang ; Qin-Yu Zhu ; Chee-Kheong Siew, "Extreme Learning Machine: A New Learning Scheme of Feed-forward Neural Networks", in IEEE International Joint Conference on Neural Networks, 12, 777-798, 2005.
15. Charles Iheagwara, Andrew Blyth, and Mukesh Singhal, "Cost Effective Management Frameworks for Intrusion Detection Systems", Journal of Computer Security, 12, 777-798, 2004.

AUTHORS PROFILE

Dr. M. S. Abirami, received M. Tech. in Computer Science and Engineering from SRM University, Chennai, India and Masters in Computer Applications from Bharathidasan University, Trichy, India. She received Ph.D. in Image Processing from Bharathiar University, Coimbatore, India. She has 19+years of teaching experience. She is currently working as Assistant Professor. Department of Software Engineering, SRM Institute of Science and Technology, Chennai, India. She has published papers 10 interpersonal journals and 12 conference papers. Her research interests include Machine Learning, Image Processing, Parallel and Distributed Computing, and Data Mining.

Shivam Pandita received B.Tech. in Software Engineering from SRM Institute of Science and Technology, Chennai, India and currently working as a system engineer specialist at Infosys. His work focuses specifically on the handling of data and data analytics. He presented a research paper in the 4th International Conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems. He is having various coding skills which include Java, Python, C and C++. His major interests are research and development in machine learning and deep learning.

Tanvi Rustagi received B.Tech. in Software Engineering from SRM Institute of Science and Technology, Chennai, India and currently working as a graduate analyst at Barclays Technology Centre, India (BTCI). Her work focuses specifically on the handling of large databases using tools such as SQL and Mongo DB. She presented a research paper in the 4th International Conference on Artificial Intelligence and Evolutionary Computations in Engineering Systems. She is having various coding skills which include Java, Python, C and C++. His major interests are research and development in machine learning and deep learning.