Improved Extreme Learning Machine (IELM) Classifier For Intrusion Detection System

R.Karthik, Dr.S.Veni, Dr.B.L.Shivakumar
Assistant Professor, Department of Information Technology, Kongunadu Arts and Science College, Coimbatore, India.
Professor & Head, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India.
Principal, Sri Ramakrishna Polytechnic College, Coimbatore, India.

Abstract- *This research work aims in design and development of an improved extreme learning machine classifier for intrusion detection system. The proposed research work contributed a single layer neural network which is trained starting with hidden nodes to the maximum number of hidden nodes and the expected learning accuracy. The improved ELM makes use of an intermediate variable in the overall recursive process which obtains better learning rate with reduced error. KDD cup'99 dataset that contains four major types of attacks in the network is chosen for performing IELM classification. Performance metrics detection rate and false alarm rate are chosen. Simulation results shows that the proposed IELM classifier outperforms in terms of improved detection rate and reduced false alarm rate.*

Keywords— *IELM, KDD cup'99, IDS, DoS,*

I. INTRODUCTION

Intrusion Detection Systems (IDSs) is an advancing modernization for guaranteeing security among computer networks. As an instance, previously denial-of-service (DoS) attack jargon carry about real disaster, though these days, productive DoS attacks can bring about overwhelming capital related hard luck to associations. The purpose of intrusion detection frameworks is to distinguish abnormal or exploitation conduct of framework and tell to network administrators about the exercises. Abundant intrusion detection setups have security limitations, for example, neglecting to encrypt the log documents, overlooking access control, and neglecting to perform trustworthiness checks, and so on. An IDS is further protected than other security gadgets, for example, firewalls [1]. Previous research works falls majorly in two significant ideas known as anomaly detection and signature detection taking into consideration anomalous conduct of the framework [2]. At first IDS comprises of accumulation of audit data from the watched framework. At that point this data is either preprocessed or specifically connected to the indicator to generate an alarm. The fundamental point of IDS is to expand detection rate and to decrease false alarm rate in recognizing attacks. As of late, the researcher for the most part centered on anomaly detection in view of proposed procedures, for example, data mining, neural system, etc.

The Intrusion detection models can be categorized into two main types: misuse-based and anomaly-based [3, 4]. A misuse-based IDS also known as signature-based or pattern-based, detecting known attacks based on information stored in a database. Although this kind of intrusion detection is efficient in detecting existing intrusions, it is fooled by any small modification in the original. Anomaly-based models can be used to detect both known and unknown intrusions, detecting deviations from normal connections [5]. The major disputes in the current anomaly intrusion detection systems are their low detection rates, which indicate that they can potentially miss detecting serious attacks and the high 'false alarm' rates, which indicate that a normal connection may be falsely classified as an attack. In general, attacks can be divided into four categories [6]:

A. Denial of Service (DoS) This type of attack is common at the scenario when an attacker intends to deny / restrict authorized users from using a Service, computer or resource. Some of the examples of DoS are SYN Flood, Ping of Death, Back, Smurf, Land, Apache2 and Teardrop [7].

B. Remote to User (R2L) Attacker seeks access to the victim machine. Examples are Send mail, Dictionary, Named, Guest, Imap, Ftp_write.

C. User to Root (U2R): Attacker with local access to the victim machine and tries to gain super user privileges. Examples are Perl, Xterm, Loadmodule, Eject, Fdformat.

D. Probing (Probe): This attack is quite common when an attacker intends to take over information through access privileges on the target host.Examples are Saint, Nmap, Mscan, Satan, and Ipsweep.

II. LITERATURE REVIEW

Decision trees also play a major role in intrusion detection [8]. The decision trees select the best features for each decision node all through the construction of the tree based on some well-defined

criteria. One such criterion is to make use of the information gain ratio, as used in C4.5. Decision trees generally have very high speed of operation and high attack detection accuracy. Debar et al. [9] and Zhang et al. [10] discuss the use of artificial neural networks for network intrusion detection. The neural networks can work effectively with noisy data; however it requires a large amount of data for training, and is often hard to decide on the best possible architecture for a neural network. Support vector machines are also used in detecting intrusions [11]. Support vector machines map real valued input feature vector to a superior dimensional feature space through nonlinear mapping and provides real-time detection capability, deals with large dimensionality of data, and is used for binary-class as well as multiclass classification. Frameworks have been proposed to overcome the weakness of single intrusion detection system [12], and they describe the collaborative use of network-based and host based systems. Maximum entropy principle [13] for detecting anomalies in the network traffic, make use of the normal data all through the training and build a baseline system. The system fails in modeling long-range dependencies in the observations.

Intelligent IDS [14, 15] achieve higher accuracy of detection with the intelligent computer programs. It investigates the environment and acts flexibly .These program compute the actions by learning the environment and by firing rules of inference[16].Intelligent IDS are capable of decision making and constraint checking .Fuzzy set [17, 18] form a key methodology for representing and processing uncertain information. Nowadays uncertainty such as imprecision, non-specificity, inconsistency, vagueness, etc arises in many forms in databases. Fuzzy sets exploit uncertainty in the attempt of making system complexity being manageable. As such, fuzzy sets represent approaches to deal with incomplete, noisy, or imprecise data. It also deals with the development of uncertain models of the data for providing smarter and smoother performance than traditional systems.

ANN approach [19] for intrusion detection is feasible in learning the new attack. The limitations of above approach were the increase in training time, and not describing why certain network traffic was intrusive. A novel multilevel hierarchical kohenen net [20] detects intrusions in networks. In their work, randomly selected data points forming KDD cup 99 were used to train and test the classifier. Their experimental observation proves that the hierarchical kohenen net in which each layer function on a small subset of the feature space was superior to kohenen net operating on the complete feature space in detecting various kinds of attacks. An IDS using NN [21] based modeling for detection of anomalous activities. The major limitation is computational load is very high. The time required for training is normally high. Most of the real life problems certainly need an optimal and acceptable solution rather than manipulating them specifically at the cost of ruined performance, time and space complexities.

SVM [22] is a supervised learning method used for solving classification and regression problems. In SVM, there is a problem called local minima.SVM can train with large number of patterns. SVM are learning machines, and plots the training vectors in high-dimensional feature space, and labels each vector by its class. It classifies the data by determining a set of support vectors, which are members of the set of the training inputs that sketch a hyper plane in the feature space. SVMs are proven as a good candidate for intrusion detection due to their speed. SVM are scalable as they are relatively insensitive to the number of data points. The classification complexity does not depend on the dimensionality of the feature space, and hence they can potentially study a larger set of patterns and scale better than neural networks. SVMs are successfully applied to many applications in the multiclass classification [23].The hybrid approach [24] combines the best results of various individual systems resulting in more accuracy. The new system is designed to have the benefits of computational efficiency and high detection accuracy in a single system.

Various research works have been carried out for intrusion detection system using support vector machines [25] – [29].

III. PROPOSED WORK

This research work aims in design and development of improved extreme learning machine classifier (IELM) for intrusion detection system. Conventional extreme learning machine is discussed in 3.1. The improved ELM part is discussed in section 3.2. for performing the classification task.

A. Extreme Learning Machine

Assume that an SLFN with *I* input neurons, *K* hidden neurons, *L* output neurons and activation function $g(\cdot)$ is trained to learn *N* distinct samples (**X**, **T**), where $X = \{x_i[n]\} \in R^{N \times 1}$ and $T = \{t_1[n]\} \in R^{N \times L}$ are the input matrix and target matrix respectively, $x_i[n]$ denotes the input data in *i*th input neuron at *n*th time instant, and $t_1[n]$ denotes the desired output in *l*th output neuron at *n*th time instant. In ELM, the input weights $\{W_{ik}\}$ and hidden

biases $\{b_k\}$ are randomly generated, where W_{ik} is the weight connecting *i*th input neuron to *k*th hidden neuron, and b_k is the bias of *k*th hidden neuron. Further let $w_{0k} = b_k \; and \; x_0[n] = 1..$ Hence, the hidden-layer output matrix $H = \{h_k[n]\} \in R^{N \times K}$ can be obtained by:

$$h_k[n] = g\left(\sum_{i=0}^{l} x_i[n].w_{ik}\right) \quad (1)$$

Let $\beta = \{\beta_{kl}\} \in R^{K \times L}$ be the matrix of output weights, where β_{kl} denotes the weight connection between k^{th} hidden neuron and l^{th} output neuron; and $Y = \{y_l[n]\} \in R^{N \times L}$ be the matrix of network output data, with $y_l[n]$ the output data in l^{th} output neuron at n^{th} time instant. Therefore, this equation can be obtained for the linear output neurons:

$$y_l[n] = \sum_{k=1}^{k} h_k[n].\beta_{kl} \quad (2)$$

$$\text{Or } Y = H.\beta \quad (3)$$

Thus, given the hidden-layer output matrix **H** and the targets matrix **T**, to minimize $\| Y - T \|_2$, the output weights can be calculated by the minimum norm least-square (LS) solution of the linear system:

$$\hat{\beta} = \mathrm{H}^{\dagger}.T, \quad (4)$$

Where $\mathrm{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix **H**. By computing output weights analytically, ELM attains good generalization performance with speedy training phase. The key step in ELM is to compute $\mathrm{H}^{\dagger}$, which generally can be done by using the singular value decomposition (SVD). Unfortunately, the computational cost of this method is dominated by the computing cost of the SVD, which is several times higher than matrix-matrix multiplication, even if a state-of-the-art implementation is used. It is revealed that if the *N* training data are distinct, **H** is full column rank; rank equals the number of columns, with probability one when $K \leq N$. In real applications, the number of hidden nodes is at all times less than the number of training data. In this case, $H^T H$ is invertible. So $\mathrm{H}^{\dagger}$ can be explicitly expressed as $(H^T H)^{-1} H^T$.

Since **H** is an *N by K* matrix (represented as $\in R^{N \times K}$), while $\mathrm{H}^{\dagger} \in R^{K \times N}$.

B. Improved ELM

The Error Minimized ELM, namely IELM, is designed to update $\mathrm{H}^{\dagger}_{k+1}$ iteratively by $\mathrm{H}^{\dagger}_{k}$, instead of H_{k+1}, when one new node is added to the existing *k* hidden nodes network. It is assumed h_{k+1} is the new column in H_{k+1} from *(k + 1)* th neuron, U_{k+1} is upper part of $\mathrm{H}^{\dagger}_{k+1}$, and is lower part of $\mathrm{H}^{\dagger}_{k+1}{}^{2}$. The key steps are:

$$\mathrm{D}_{k+1} = \frac{\mathrm{h}^{\mathrm{T}}_{k+1}(\mathrm{I} - \mathrm{H}_k \mathrm{H}^{\dagger}_{K})}{\mathrm{h}^{\mathrm{T}}_{k+1}(\mathrm{I} - \mathrm{H}_k \mathrm{H}^{\dagger}_{K})\mathrm{h}_{k+1}} \quad (5)$$

$$\mathrm{U}^{k+1} = \mathrm{H}^{\dagger}_{K}(\mathrm{I} - \mathrm{h}_{k+1}\mathrm{D}_k) \quad (6)$$

$$\mathrm{H}^{\dagger}_{k+1} = \begin{bmatrix} \mathrm{U}_{k+1} \\ \mathrm{D}_{k+1} \end{bmatrix} \quad (7)$$

Although it is claimed that the training time of IELM is less than that of ELM, it seems not true with simple analysis if the above formula is used directly. The most computational consuming step of IELM is multiplication of H_k and $\mathrm{H}^{\dagger}_{k}$, with complexity $\mathrm{O}(\mathrm{kN}^2)$, even more than $\mathrm{O}(\mathrm{k}^2\mathrm{N})$ in ELM. (Note $K << N$ is usually the case.) However, with a little modification in EM-ELM, we can decrease the computational complexity considerably, with complexity of $\mathrm{O}(5kN + 2N)$ when even updating the output weights ***β***:

$$\mathrm{D}^{\dagger}_{k+1} = \frac{h^t_{k+1} - h^t_{k+1} H_k \mathrm{H}^{\dagger}_{k}}{h^t_{k+1} h_{k+1} - h^t_{k+1} H_k \mathrm{H}^{\dagger}_{k} \mathrm{h}_{k+1}} \quad (8)$$

$$U_{K+1} = \mathrm{H}^{\dagger}_{k} - \mathrm{H}^{\dagger}_{k} \mathrm{h}_{k+1} \mathrm{D}_k \quad (9)$$

$$\beta_{K+1} = \begin{bmatrix} U_{K+1} \\ D_{K+1} \end{bmatrix} T \quad (10)$$

Given a set of training data, assume that a single hidden layer neural network is to be trained, starting with 1 hidden nodes, to maximum number of hidden nodes K_{max}, and the expected learning accuracy $\in$. Note that it is R^{-1}_{k} instead of R_k used as an intermediate variable in the whole recursive process,

hence $P_k = R^{-1}{}_k$ is introduced in the procedure. The whole process of IELM is depicted in the Algorithm1.

1: Randomly generate the single hidden node input weights set $\{\omega_{i1}\}^I{}_{i=0}$

2: Calculate the hidden-layer output matrix $H_1(=h_1)$

3: Calculate the inverse of

$$R_1 : P_1(=p_{11}=\frac{1}{r_{11}}) = (h^T{}_1 h_1)^{-\frac{1}{2}}$$

4: Calculate $Q_1 = q_1 = p_{11} h_1$

5: Calculate the output weight $\hat{\beta}_1(=\beta_1) = P_1 Q^T{}_1 T$

6: while $k = 1$ to K_{max} and $E(H_k) = \| H_k \beta_k - T \| > \epsilon$ do

7: A new hidden node is added, the corresponding input weights set are generated $\{\omega^i{}_{,k+1}\}^I{}_{i=0}$ i=0, and the corresponding h_{k+1} are calculated.

8: Update the following variables in sequence:

$$\delta r_{k+1} = Q^T{}_k h_{k+1} \quad (11)$$

$$\delta h_{k+1} = h_{k+1} - Q_k \delta r_{k+1} \quad (12)$$

$$p_{k+1,k+1}\left(=\frac{1}{r_{k+1,k+1}}\right) = (\delta h^T{}_{k+1,k+1} \delta h_{k+1,k+1})^{-\frac{1}{2}} \quad (13)$$

$$q_{k+1} = p_{k+1,k+1} \delta h_{k+1,k+1} \quad (14)$$

$$\beta_{k+1} = p_{k+1,k+1} q^T{}_{k+1} T \quad (15)$$

$$\beta_{k+1} = \begin{bmatrix} \hat{\beta}_K - P_K \delta r_{K+1} \beta_{K+1} \\ \beta_{k+1} \end{bmatrix} \quad (16)$$

$$p_{k+1} = \begin{bmatrix} P_K \quad | - P_k \delta r_{K+1} p_{k+1,k+1} \\ 0 | \; p_{k+1,k+1} \end{bmatrix} \quad (17)$$

$$Q_{k+1} = [Q_k \mid q_{k+1}] \quad (18)$$

9: $k \leftarrow k+1$

10: while end

IV. EXPERIMENTS

The KDD Cup 1999 dataset used as in [28], [29] is used for benchmarking intrusion detection problems in our research work. The dataset was a collection of simulated raw TCP dump data over a period of nine weeks on a local area Network. The training data was processed to about five million connections records from seven weeks of network traffic and two weeks of testing data received around two million connection records. The training data is made up of 22 different attacks out of the 39 in the test data. The known attack types are those present in the training dataset while the novel attacks are the added attacks in the test datasets not available in the training data sets. The attacks types are grouped into four categories:

DOS: Denial of service – e.g. syn flooding

Probing: Surveillance and other probing, e.g. port scanning

U2R: unauthorized access to local super user (root) privileges, e.g. buffer overflow attacks.

R2L: unauthorized access from a remote machine like password guessing

The training dataset consisted of 4, 94,021 records among which 97,277 (19.69%) were normal, 3,91,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R connections. In each connection, 41 attributes describes the different features of the connection and a label assigned to each either as an attack type or as normal. Simulation results shows that the proposed IELM attains better detection rate and reduced false alarm rate.

	DoS	Probe	U2R	R2L
IELM	98.1	98.7	97.7	63
ELMwith Semantic Feature [27]	96.8	75	5.3	4.2
PSO with SVM [26]	97.9	98.6	68.9	19.5

Table 1 Detection Rate

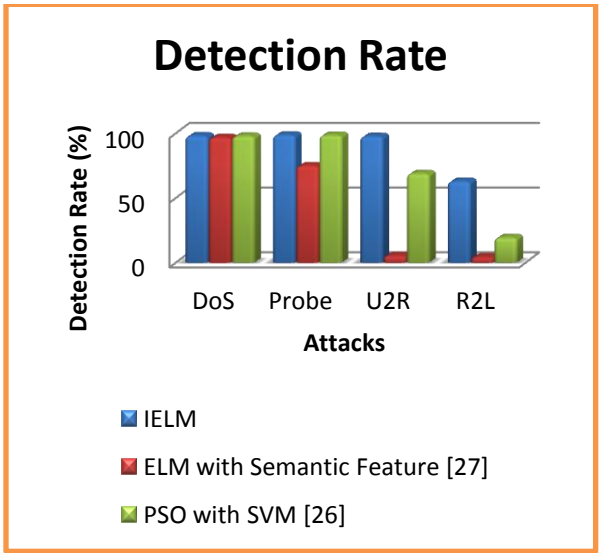

Fig 1 Detection Rate

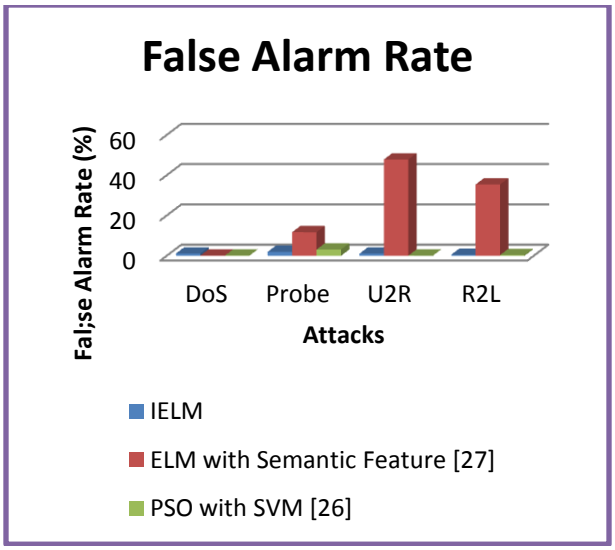

Fig 2 False Alarm Rate

	DoS	Probe	U2R	R2L
IELM	1.09	1.95	1.02	0.5
ELM with Semantic Feature [27]	0.1	11.7	47.8	35.4
PSO with SVM [26]	0.07	3.1	0.05	0.35

Table 2 False Alarm Rate

V. CONCLUSIONS

The proposed research work intends in design and development of improved extreme learning machine classifier (IELM) for intrusion detection system. The proposed research work contributed a single layer neural network which is trained starting with hidden nodes to the maximum number of hidden nodes and the expected learning accuracy. The improved ELM makes use of an intermediate variable in the overall recursive process which obtains better learning rate with reduced error. KDD cup'99 dataset that contains four major types of attacks in the network is chosen for performing IELM classification. Performance metrics detection rate and false alarm rate are chosen. Simulation results shows that the proposed IELM classifier outperforms in terms of improved detection rate and reduced false alarm rate.

REFERENCES

[1] Overview of Attack Trends, attack_trends.pdf, 2002.

[2] K.K. Gupta, B. Nath, R. Kotagiri, and A. Kazi, "Attacking Confidentiality: An Agent Based Approach," Proc. IEEE Int'l Conf. Intelligence and Security Informatics (ISI '06), vol. 3975, 2006, pp. 285-296.

[3] J Anderson, An Introduction to Neural Networks (MIT, Cambridge, 1995)

[4] B Rhodes, J Mahaffey, J Cannady, Multiple self-organizing maps for intrusion detection, Paper presented at the Proceedings of the 23rd National Information Systems Security Conference, Baltimore,2000, pp 16–19.

[5] A. Sung, S. Mukkamala, Identifying important features for intrusion detection using support vector machines and neural networks in *Symposium on Applications and the Internet*, 2003, pp. 209–216.

[6] R.Karthik, and B.L Shivakumar, "A Taxonomy of Network Intrusion Detection System for Wireless Communication", International Journal of Computer Science And Engineering, vol. 3, December 2015, pp. 35-42, issue-12, E- ISSN: 2347-2693.

[7] R.Karthik, S.Veni and B.L Shivakumar, "Fuzzy Based Support Vector Machine Classifier With Wiener Filter (Fsvm – Wf) For Intrusion Detection System", International Journal of Advanced Research in Computer Science , vol. 7, July – August 2016, pp. 11-15, issue-4, E- ISSN: 0976-5697.

[8] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), 2000, pp. 420-424.

[9] H. Debar, M. Becke, and D. Siboni, "A Neural Network Component for an Intrusion Detection System," Proc. IEEE Symp. Research in Security and Privacy (RSP '92), 1992, pp. 240-250.

[10] Z. Zhang, J. Li, C.N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: A Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," Proc. IEEE Workshop Information Assurance and Security (IAW '01), 2001, pp. 85-90.

[11] D.S. Kim and J.S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Proc. Information Networking, NetworkingTechnologies for Enhanced Internet Services Int'l Conf. (ICOIN '03), 2003, pp. 747-756.

[12] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi, "Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC '03), 2003, pp. 234-244.

[13] Y. Gu, A. McCallum, and D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," Proc. Internet Measurement Conf. (IMC '05), 2005, pp. 345-350, USENIX Assoc.

[14] S Franklin, A Graser, Is it an agent or just a program? in ECAI '96 Proceedingsof the Workshop on Intelligent Agents III, Agent Theories, Architectures, andLanguages (Springer, London, 1996)

[15] N Jaisankar, SGP Yogesh, A Kannan, K Anand, Intelligent Agent BasedIntrusion Detection System Using Fuzzy Rough Set Based Outlier Detection, SoftComputing Techniques in Vision Sci., SCI 395 (Springer, 2012), pp. 147–153.

[16] T Magedanz, K Rothermel, S Krause, Intelligent agents: an emergingtechnology for next generation telecommunications? In INFOCOM'96Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computerand Communications Societies, San Francisco, 1996 Mar 24–28.

[17] W Zhang, S Teng, H Zhu, H Du, X Li, Fuzzy Multi-Class Support VectorMachines for Cooperative Network Intrusion detection. Proc. 9th IEEE Int.Conference on Cognitive Informatics (ICCI'10) (IEEE, Piscataway, 2010), pp. 811–818.

[18] L Zadeh, Role of soft computing and fuzzy logic in the conception, design and development of information/intelligent systems, in Computational Intelligence: Soft Computing and Fuzzy-neuro Integration with Applications, ed.by O Kaynak, L Zadeh,

B Turksen, I Rudas. Proceedings of the NATO Advanced Study Institute on Soft Computing and its Applications held at Manavgat, Antalya, Turkey, 21–31 August 1996, volume 162 of NATO ASISeries (Springer, Berlin, 1998), pp. 1–9.

[19] M Moradi, M Zulkernine, A neural network based system for intrusiondetection and classification of attacks, in Proceedings of IEEE InternationalConference on Advances in Intelligent Systems – Theory and Applications,Luxembourg, vol. 148 (IEEE, Amsterdam, 2004), pp. 1–6

[20] S Sarasamma, Q Zhu, J Huff, Hierarchical Kohonen net for anomalydetection in network security. IEEE Transactions on System, Man,Cybernetics, Part B, Cybernetics 35(2), 302–312 (2005)

[21] O Linda, T Vollmer, M Manic, Neural network based intrusion detectionsystem for critical infrastructures, in Proceedings of IEEE International JointConference on Neural Networks, Georgia (IEEE, Amsterdam, 2009), pp. 102–109

[22] C Cortes, V Vapnik, Support vector networks. Mach. Learn. 20, 1–25 (1995)

[23] S.-W. Lin, K.-C. Ying, C.-Y. Lee and Z.-J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection", Applied Soft Computing, vol. 12, **(2012)**, pp. 3285-3290.

[24] VahidGolmah, An Efficient Hybrid Intrusion Detection System based on C5.0 and SVM, International Journal of Database Theory and Application Vol.7, No.2 (2014), pp.59-70.

[25] Zhai Jinbiao, Research on Intrusion Detection System Based on Clustering Fuzzy Support Vector Machine, International Journal of Security and Its Applications Vol.8, No.3 (2014), pp. 249-260.

[26] Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal, Shamim Ahmad, Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS), Journal of Intelligent Learning Systems and Applications, 2014, 6, 45-52.

[27] Jashan Koshal, Monark Bag, Cascading of C4.5 Decision Tree and Support Vector Machine for Rule Based Intrusion Detection System, I. J. Computer Network and Information Security, 2012, 8, 8-20.

[28] B. Sujitha, V. Kavitha, "Layered Approach For Intrusion Detection Using Multiobjective Particle Swarm Optimization", International Journal of Applied Engineering Research, vol.10, no.12, pp. 31999 – 32014, 2015.

[29] G. Creech, J. Hu, "A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Discontiguous System Call Patterns", IEEE Transactions on Computers, vol. 63, no. 4, pp. 807 – 819, 2014.