A hybrid Particle swarm optimization -Extreme Learning Machine approach for Intrusion Detection System

Mohammed Hasan Ali[1,2], Mohamad Fadlizolkipi[3], Ahmad Firdaus[3],
Nik Zulkarnaen Khidzir[4]
[1,2,3] Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Pahang
[4]Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan,
Malaysia
Mh180250@gmail.com

***Abstract*—There are several limitations that facing intrusion-detection system in current days, such as high rates of false positive alerts, low detection rates of rare but dangerous attacks. Daily, there are reports of incidents such as major ex-filtration of data for the purposes of stealing identities. Hybrid model's approaches have been widely used to increase the effectiveness of intrusion-detection platforms. This work proposes the extreme learning machine (ELM) is one of the poplar machine learning algorithms which, easy to implement with excellent learning performance characteristics. However, the internal power parameters (weight and basis) of ELM are initialized at random, causing the algorithm to be unstable. The Particle swarm optimization (PSO) is a well-known meta-heuristic which is used in this research to optimize the ELM. Our propose model has been apple based as intrusion detection and validated based on NSL-KDD data set. Our developed model has been compared against a basic ELM. PSO-ELM has outperformed a basic model in the testing accuracy**

Keywords—intrusion detection system; extreme learning machine; Particle swarm optimization; NSL-KDD; hybrid

I. INTRODUCTION

People have over the years depended on technology and computer networks for their daily activities [1-2], such as messaging, shopping, and marketing. These networks are constantly exposed to several online threats and for this reason, their integrity and availability ought to be protected against violation and intrusion. Inexperienced intruders, terrorist organizations, and rival corporations have the motive and capability to carry out sophisticated attacks against computer systems [3-4], and this makes systems security an important issue for many researchers. Network intrusion or attack is a situation where an unauthorized user (attacker) tries to exploit the vulnerability of a system to gain access to network resources and cause a disruption in the normal operation of the network[5-6]. If the attacker gains access to the network resources, it can have an illegitimate access to the network data, can modify or corrupt the system, and can hijack the network or alter its behavior.

Since Denning introduced the concept of detecting intrusion detection [7], many efforts have been channeled on the ability for network monitoring tools to automatically detect network intrusion. An intrusion detection system (IDS)[5][8] is a system that automatically monitors network or system activities, analyzes them for any sign of intrusion, and often prevents unauthorized network access [4][9].

Therefore, requires a significant level of adjustment and tuning by human operators. This has resulted in systems that are not reliable or effective in real-world network operational environments. As new attacks are invented regularly, it is not sufficient to rely on classical signature-based ID. This has motivated studies that focus on the anomaly-based machine learning for IDS [10]. The performance of IDS is improved by the incorporation of machine learning (ML) [11-12], ML algorithms can theoretically achieve optimum performance, i.e. it can regulate the rate of false alarms and improve the detection accuracy.

To improve the performance of ML algorithms in terms of detection accuracy performance, several works proposed hybrid with optimization algorithms[3][13][16]. Moreover, hybrid models improved intrusion detection also in several works [10][17] and have shown better results in compared with IDS based single algorithm. In this work, particle swarm optimization algorithm is used to select the main parameters to reduce the impact of randomization on the IDS based extreme learning machine(ELM) as a core algorithm for the proposed IDS models. The present work remainder is arranged as such: we start in Section2. Represents overview of algorithms (ELM, PSO) Section.3 include developed of methodology. Section.4. showed the discussion and result. The conclusion of this work represents in section.5.

II. OVERVIEW OF ALGORITHMS

A. Overview of Extreme Learning Machine

ELM was first proposed by Huang and has been widely studied by researchers. It has also been applied to a variety of applications [18]. ELMs are structured as single layer feed-forward neural networks which analytically determine their output weights rather than using an iterative tuning algorithm like backpropagation in the case of a standard neural network or solving a quadratic programming optimization problem as is the case with an SVM [19]. The parameters of the hidden node in an ELM are either chosen randomly or set up as a kernel function. ELMs rely on the computation of a matrix inverse to determine a mapping from the input data to the target function and in the case of the kernel based ELM, a Kernel function is used in place of the random parameter

978-1-5386-9175-5/18/$31.00 ©2018 IEEE

selection. Since ELMs does not require iterative processing, they tend to run much faster than their neural network (NN) or support vector machine (SVM) counterparts.
The use of ELMs instead of SVMs is mainly due to the ability of ELMs to achieve comparable classification accuracy rates, as well as its lower computational requirements compared to SVMs [19]. Despite the proven success of SVM when applied to several ML problems, the ELM has shown more advantages compared to the SVM.

ELMs are developed with inbuilt multiclass classification capability. Its results are calculated in a similar manner irrespective of the number of classes contained in the learning problem. This conferred ELMs with several advantages over SVMs for most real-world usage[19]. The ELMs are basically structured from SLFN and familiar to those previously exposed to NN [20]. An SLFN is composed of an output layer that corresponds to the learning decision and a hidden layer of computational nodes . The basic structure of SLFN is depicted in Fig. 1.

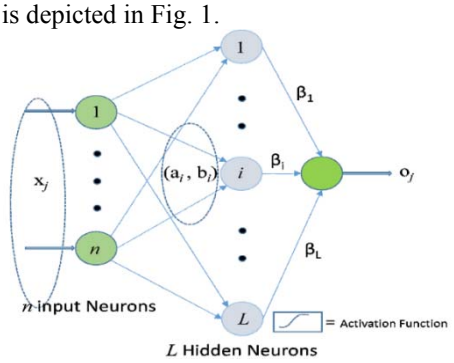

Fig. 1. Basic structure of SLFN

Huang has shown that the basic structure of SLFN depicted in Fig. 1, the number of hidden neurons can increase as much as required for provide decisions of classification for any disjoint regions. The SLFN depicted in Fig. and output function as following:

$$f_{L}(\vec{x}) = \sum_{i=1}^{L} G(\vec{x}, a_i, b_i)\,\beta_i \qquad (1)$$

Where L is the number if nodes contained in the hidden layer, and x the input data. A is the weights or connection strength between each of the i hidden node and the inputs, which b represents a hidden node threshold. G represents the activation function for expressing the additive nodes as g(a.x+b). The activation function for the RBF nodes is defined as $g(a.x+b) = g(b_i\|x - a_i\|) = e^{-b_i\|x - a_i\|^2}$ Systems for classification problems should map the input data set x based on N examples via the transformation of G into m classes t. By defining the mapping function h(x) then the equation can then be expresses thus:

$$H\beta = T \qquad (2)$$

Where H is represents the output matrix of the hidden layer between T and x as [10] In this case:

$$\boldsymbol{H} = \begin{bmatrix} g(a_1.x_1 + b_1) & \cdots & g(a_L.x_1 + b_L) \\ \vdots & \vdots & \vdots \\ g(a_1.x_N + b_1) & \cdots & g(a_L.x_N + b_L) \end{bmatrix}_{N\times L} \qquad (3)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L\times m} \qquad (4)$$

$$\boldsymbol{T} = \begin{bmatrix} t_{1m}^T \\ \vdots \\ t_{Nm}^T \end{bmatrix}_{N\times m} \qquad (5)$$

Given, above equations , then the learning objective will be obtaining an a best solution β ,S which provide less training error during classify new data sets with similar features to the training set. This implies the ability of the system to generalize when faced with systems with new data.

B. Overview of Particle swarm optimization

Particle swarm optimization (PSO) is a parallel evolutionary developed by kennedy and eberhart [21]. The feature of the PSO as compared to the evolution algorithms such as Genetic algorithm (GA) lies in its easy in implementation, and its necessitate just a few parameters. [22], [23]. In the D-dimensional space, the PSO is denoted his ith particle as $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ where $x_{id} \in [LB_d, UB_d]$, d $\in [1, D]$, LB_d, UB_d respectively represents the minimum and maximum limits of the dth dimension. The velocity of particle i is given as $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots\ldots, v_{iD})$, Which is maintained at maximum velocity that specified by a user V_{max}. The particles, at each time step t, are manipulated based on the following relation:

$$v_i(k+1) = w v_i k + c1r1\,(xbest,\ local - x_i) + c2r2\,(xbest,\ global - x_i) \qquad (6)$$

$$x_i(k+1) = xk + v(k+1) \qquad (7)$$

The velocity and position of each particle are represented as the vectors $v_k = (v_{k1}, \ldots, v_{kd})$ and $x_i = (x_{i1}, \ldots, x_{id})$ respectively. In (6) x vectors are representing the best local and best global positions. $c1$ and $c2$ are acceleration factors known as cognitive and social parameters. $r1$ and $r2$ are random number between 0 and 1. k is the iteration index. w is the inertia weight parameter[24]. And update x_i for particle using (7).

III. Proposed Methodology

AS mentioned in previous parts that ELM algorithm randomly selects the weight values and its distribution. Moreover, will be degraded performance unless an appropriate way in order to select the best weights. Our work proposes PSO-Based optimized ELM is trained to selecting weights.

PSO-ELM designing based on a PSO particle represents weights of ELM. In addition, to accomplish a better accuracy the optimization is requiring to select both the weight's values as well as the number of neurons that are needed in the hidden layer. Moreover, sigmoid is used as activation function for output of the hidden layer neurons. Also, PSO parameters has been set with 100 number of iteration, and 50 number of of particles and other parameters such as c_1, c_2,w adjusted as standard version [21].The PSO-based optimization pseudo-code is shown as following.

1. Create Initial Generation of Particles.$P_i = \{w_j\}$, i =

1,…N J= 1, …M, M weights number ,N population size.

2. For each particle do the following
3. For each particle build an equivalent ELM network.
4. Do the following for each ELM
5. 1) Calculate ELM accuracy
6. 2) If the value of fitness is better than the best local fitness value (pLBest) in history
7. The current values set as the pLBest
8. End If
9. 3) If the best global fitness value is better than the(pLBest) fitness value set current
10. value as the new pLBest
11. Update particle position according to the position equation
12. Go to 4

IV. Results of Expremental

In order to validate the PSO-ELM proposed learning model, heavy comparison with basic ELM has been prepared with a different number of neurons in the hidden layer. The NSL-KDD data set was used to evaluate the performance of proposed models. The NSL_KDD data set consists of several types of features such as symbolic, numeric and Boolean with varying resolution and ranges. For this study we used all 41 attributes of the data.

As of NSL-KDD dataset consist of 148517 number of connection records in both training and testing set with same number of attributes. The data was divided into 80% for training and 20% for testing based cross-validation technique. All the results in the following table represents as average for runs fifteen times. The proposed model PSO-ELM compared with basic ELM based on different number of neurons in the hidden layer to measure the effect the diffraction of the number of the neurons on the model's accuracy. As following table.1. shown the results based different standards evaluation includes max accuracy (Max.Acc), Average Accuracy (Avr.ACC), Detection Rate(DR), False Alarm Rate (FAR), Precision, Recall, F-measure (F.M), and G-mean (F.M).

TABLE I. Results of compaction

A.

No.Neurons	Model	Max.Acc	Avr.Acc	DR	FAR
10	ELM	0.9255	0.8956	0.9047	0.1545
	PSO-ELM	0.949	0.9388	0.9285	0.0822
25	ELM	0.9521	0.9471	0.9418	0.0695
	PSO-ELM	0.9725	0.9687	0.9674	0.0415
35	ELM	0.9631	0.9548	0.9501	0.0591
	PSO-ELM	0.9817	0.9791	0.9759	0.0211
50	ELM	0.9709	0.9652	0.9593	0.0478
	PSO-ELM	0.9864	0.9845	0.9806	0.0132

B.

No.Neurons	Model	Precision	Recall	F.M	G.M
10	ELM	0.8956	0.8955	0.8955	0.8061
	PSO-ELM	0.9182	0.9165	0.9117	0.8834
25	ELM	0.9469	0.9472	0.9471	0.8977
	PSO-ELM	0.9689	0.9676	0.9656	0.9163
35	ELM	0.9632	0.9628	0.9629	0.9124
	PSO-ELM	0.9736	0.9774	0.9745	0.9233
50	ELM	0.9706	0.9701	0.9709	0.9321
	PSO-ELM	0.9833	0.9856	0.9864	0.9564

In both parts A and B showed how PSO-ELM model improved basic ELM as it's achieved better accuracy and that is showed the impact reduce randomized for select the main ELM parameters.

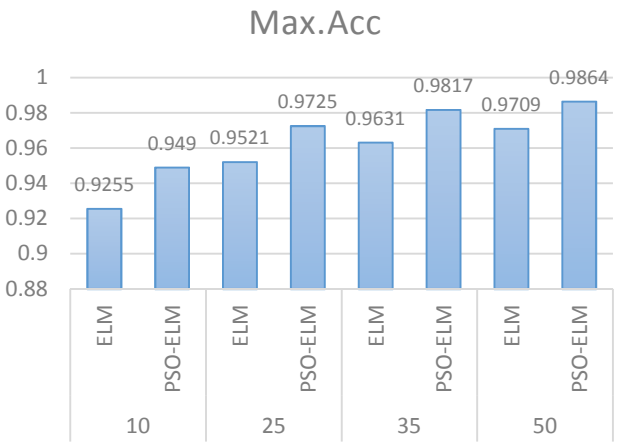

Fig. 2. Accuracy compartion (PSO-ELM Vs ELM)

Moreover, the table showed proposed model PSO-ELM with less number of neurons which me less complexity achieved better accuracy. Fig. 2. showed accuracy compared between PSO-ELM and ELM based different number of neurons.

V. Conclusion

Intrusion-detection system represents one of the important research areas for researchers. On another hand, IDS face different limitations such as false alarm and low detection rate. This work proposed hybrid model based optimization ELM by PSO to provide the main ELM parameters instead of selected randomly. The NSL-KDD data set was used to evaluate the performance of proposed models. The results showed the proposed model PSO-ELM improved the accuracy with fewer numbers of neurons in compared with basic ELM. The future work to evaluate the proposed model with different data sets and to optimize the core algorithm ELM with the different optimization algorithms to achieve better results in compare with current situation

Acknowledgment

This work was supported by Universiti Malaysia Pahang, under the Grant IBM Centre of Excellence (COE)(IBM2000), RDU180337.

REFERENCES

[1] M. H. Ali1, "TOWARDS A EXCEPTIONAL DISTRIBUTED DATABASE MODEL FOR MULTI DBMS." pp. 553–560, 2014.

[2] H. A. S. Ahmed, M. H. Ali, L. M. Kadhum, M. Fadli, B. Zolkipli, and Y. A. Alsariera, "A Review of Challenges and Security Risks of Cloud Computing," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 1, pp. 87–91, 2016.

[3] M. M. Ali, Mohammed Hasan, "Optimize Machine Learning Based Intrusion Detection for Cloud Computing : Review Paper," *J. Eng. Appl. Sci.*, vol. 11, no. Special Issue 2, pp. 3254–3264, 2016.

[4] M. H. Ali and M. F. Zolkipli, "Review on Hybrid Extreme Learning Machine and Genetic Algorithm To Work As Intrusion Detection System in Cloud Computing," vol. 11, no. 1, pp. 460–464, 2016.

[5] M. H. Ali, M. F. Zolkipli, M. A. Mohammed, and M. M. Jaber, "Enhance of extreme learning machine-genetic algorithm hybrid based on intrusion detection system," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4180–4185, 2017.

[6] N. B. A. G. and M. A. M. Mohammed Hasan Ali , Mohamad Fadli Zolkipli, "Intrusion Detection System Based on Machine Learning in Cloud computing," *J. Eng. Appl. Sci.*, vol. 11, no. 14, pp. 3279–3284, 2016.

[7] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, no. 2, pp. 222–232, 1987.

[8] M. H. Ali, M. F. Zolkipli, M. M. Jaber, and M. A. Mohammed, "Intrusion detection system based on machine learning in cloud computing," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4241–4245, 2017.

[9] K. Scarfone, P. Mell, and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology."

[10] J. M. Fossaceca, T. A. Mazzuchi, and S. Sarkani, "MARK-ELM: Application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection," *Expert Syst. Appl.*, vol. 42, no. 8, pp. 4062–4080, 2015.

[11] M. H. Ali and M. F. Zolkipli, "Model of Improved a Kernel Fast Learning Network Based on Intrusion Detection System," pp. 1–12, 2019.

[12] M. H. Ali and M. F. Zolkipli, "Intrusion-Detection System Based on Fast Learning Network in Cloud Computing," no. October, 2018.

[13] A.-C. Enache and V. Sgarciu, "Anomaly intrusions detection based on support vector machines with bat algorithm," *Syst. Theory, Control Comput. (ICSTCC), 2014 18th Int. Conf.*, pp. 856–861, 2014.

[14] P. Niu, K. Chen, Y. Ma, X. Li, A. Liu, and G. Li, "Model turbine heat rate by fast learning network with tuning based on ameliorated krill herd algorithm," *Knowledge-Based Syst.*, vol. 118, pp. 80–92, 2017.

[15] Z. Cao *et al.*, "Optimization of gear blank preforms based on a new R-GPLVM model utilizing GA-ELM," *Knowledge-Based Syst.*, vol. 83, no. 1, pp. 66–80, 2015.

[16] M. H. Ali, B. A. D. AL Mohammed, M. A. B. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on Fast Learning Network and Particle swarm optimization," *IEEE Access*, vol. XX, no. c, pp. 1–1, 2018.

[17] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Comput. Networks*, vol. 136, pp. 37–50, 2018.

[18] N. Computing, M. Technology, M. Technology, and M. Technology, "Extreme learning machine and its applications," no. August 2014, 2013.

[19] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. man, Cybern. Part B, Cybern.*, vol. 42, no. 2, pp. 513–29, 2012.

[20] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6225–6232, 2010.

[21] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int. Conf. Neural Networks 1995*, vol. 4, pp. 1942–1948, 2011.

[22] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Inf. Process. Lett.*, vol. 93, no. 5, pp. 255–261, 2005.

[23] S. C. Satapathy, B. N. Biswal, S. K. Udgata, and J. K. Mandal, "Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014," *Adv. Intell. Syst. Comput.*, vol. 327, no. January, 2014.

[24] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," *1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell. (Cat. No.98TH8360)*, pp. 69–73, 1998.