Propose a new Firefly-Fast Learning Network model based Intrusion-Detection System

Mohammed Falih Badran, Kohbalan Moorthy, Nor Saradatul Akmar Zukifi, Mohd Saberi Mohamad, Safaai Deris, Nan Md.Sahar

Abstract— Currently, effective Intrusion-detection systems (IDS) still represent one of the important security tools. However, hybrid models based on the IDS achieve better results compared with intrusion detection based on a single algorithm. But even so, the hybrid models based on traditional algorithms still face different limitations. This work is focused on providing two main goals; firstly, analysis based on the main methods and limitations of the most-recent hybrid model-based on intrusion detection, secondly, to propose a novel hybrid IDS model called FA-FLN based on the Firefly algorithm and Fast Learning Network.

Keywords: Fast Learning Network, Intrusion Detection System, Optimization

I. INTRODUCTION

Technology has over the many years impacted the current days based on several applications like marketing, shopping, and messaging [1]. A major problem is that these networks are steadily exposed to numerous online threats which threaten their availability and integrity and as such, demands to be protected from intrusion and violation. In 2015, the U.S. Director of NSA, Adm. Michael Rogers, in the House Intelligence Committee, warned of an impending major security attack in the U.S. in the next decade. In his words, "It's only a matter of the 'when,' not 'if,' that we are going to see something dramatic." Several state-backed hackers have continuously launched attacks on industrial control systems that manage vital infrastructures, such as nuclear power, power grid, transportation systems, and air-traffic control. The NSA director also opined that, based on his own assessment, the U.S. may fall into these attacks [2].

Furthermore, Intrusion Detection System (IDS) is one of the powerful software or hardware [3] that is used to monitor computer network for the detection of normal or abnormal behaviors [4][5]. An IDS monitors a network for signs of invasion which could manifest in abnormal system behaviors or violation of network security policies. Moreover, there are several limitations of the conventional IDS [6], [7], such as high rate false alarms, lack of continuous adaptation to changing malicious behaviors, and highly uneven data distribution. Furthermore, the incorporation of machine learning (ML) can enhance the performance of IDS [8], [9] as the ML algorithms can ensure optimum performance. This work provides several contributions based on ML models: firstly, analysis of the most recent models of ML-based IDS, secondly, proposed a new hybrid model which includes Fast Learning Network (FLN) and Firefly (FA) algorithms which can fill the gaps in the current ML models based on IDS.

II. OVERVIEW OF INTRUSION DETECTION SYSTEM

Technological advancements in the present world have made connectivity easier than ever [10]. A large amount of information (personal, military, government, and commercial) are hosted on network infrastructures worldwide. The security of network infrastructures is attracting great research interest due to the huge number of intellectual properties which can be easily acquired through the internet. The society has become over-reliant on technology as people depend on computer systems for their daily information and entertainment [11].

Moreover, IDS represents one of powerful security tool which monitoring the system activities for any abnormal system behaviors or violation of network security policies. Moreover, IDS perform several functions [12] such as Monitors and analyzes the activity of the system users and Checks the critical system and data file integrity. In general IDS techniques divided into anomalies or signatures of attack are used by the detection system for the detection of attacks, and these techniques determine the effectiveness of an IDS [9], [13] in following Table.1 represents the main difference between IDS techniques.

Table.1 comparison between Anomaly and signature detection

Aspects	Anomaly Detection	Signature
Characteristics	Uses the deviation from normal usage patterns to identify intrusions.	Identifies intrusion using know attack signatures.
Drawbacks	Must study the sequential interrelation between transactions, False positives.	Known attacks must be coded manually; cannot detect new attacks, signatures must be regularly updated

Revised Manuscript Received on September 14, 2019.

Mohammed Falih Badran, Microelectronics and Nanotechnology Shamsudin Research Center(MiNT-SRC), Faculty of Electrical and Electronics Engineering, Hussein Onn University Malaysia

Kohbalan Moorthy, Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

Nor Saradatul Akmar Zukifi, Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Kuantan, Malaysia

Mohd Saberi Mohamad, Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, Kota Bharu, Malaysia

Safaai Deris, Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, Kota Bharu, Malaysia

Nan Md.Sahar, Microelectronics and Nanotechnology Shamsudin Research Center(MiNT-SRC), Faculty of Electrical and Electronics Engineering, Hussein Onn University Malaysia

Furthermore, works proposed hybrid models based on IDS such as optimize for machine learning algorithms achieved better results in compared with models based on single algorithm or method such as [14],[15],[16][17]–[20]. However, most of these hybrid models based on IDS still facing several limitations because most of these model includes old algorithms, still manually part involved in the proposed model structure. In the following section, this work provides an analysis for most of the recent IDSs based on hybrid models.

III. OVERVIEW OF IDS BASED ON MACHINE LEARNING

The conventional techniques like firewalls, encryption, and access control have been proven inefficient in adequately protecting networks from the ever-increasingly forms of attacks and malware [12]. Consequently, the IDS have been developed as an indispensable aspect of security systems which is used for the detection of attacks even before they occur [21] [22]. There are certain issues to consider when building IDS, issues like data collection, intrusion recognition, data pre-processing, reporting, and response. The most important among these issues is intrusion recognition.

Similarly, ML has not been so good in terms of processing time and accuracy when faced with these demands [23]. Fortunately, the ability of computational intelligence techniques to exhibit fault tolerance, coupled with their high computational speed and robustness to noisy data have compensated for these drawbacks. Most of the ML-based systems are susceptible to high false positive and false-negative alarm rates. They also lack the ability to continuously adapt to emerging attack behaviors [24]. To overcome most of these ML limitations, several optimization techniques have been merged with machine learning algorithms. Among these techniques include Genetic Algorithm (GA), Bees Algorithm, and Particle Swarm Optimization (PSO). This work will analyze most of the recent IDSs that have been proposed based on hybrid models.

IV. MAIN STRUCTURE OF PROPOSED MODEL

This section represents explain for the propose a model which includes, basic Firefly and Fast Learning Network algorithms. Moreover, this section includes steps of the new model (FA-FLN) based on IDS.

4.1 Overview of Fast Learning Network

The FLN is comprised of a single layer feedforward neural network (FFNN) parallelly connected with a three-layer FFNN that consists of input, hidden, and output layers [25]. Figure 1 depicts the structure of FLN. Assume a set of N arbitrary discrete samples $\{(x_i, y_i), i = 1, 2, \ldots, N\}$ with $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ being the n-dimensional eigenvector of the i^{th} sample, and $y_i = [y_{i1}, y_{i2}, \ldots, y_{iI}]^T \in R^I$ being the associated I-dimension output vector. Let *m* represent the number of nodes in the hidden layer (the number of neurons in the hidden layer can be determined using different methods). For instance, it can be determined by setting the number of hidden neurons in between the size of the input and output layers [26]. The active function of the hidden nodes is represented by $\gamma(.)$ [27]. FLN can be modeled mathematically using the provided vectors and matrices as in the equations:

$$y_j = f\,(w^{oi}x_j + \sum_{k=1}^{m} w_k^{oh}\, g\,(w_k^{in}x_j + b_k)) \qquad 2(1)$$

Where j=1,2,….,N, $w^{oi} = [w_1^{oi}, w_2^{oi}, \ldots\ldots, w_I^{oi}]$ represent the weight vector that connects the j^{th} input and output nodes, $w_k^{in} = [w_{k1}^{in}, w_{k2}^{in}, \ldots\ldots, w_{km}^{in}]$ represents the weight vector that connects the k^{th} input and hidden nodes, $w_k^{oh} = [w_{1k}^{oh}, w_{2k}^{oh}, \ldots\ldots, w_{Ik}^{oh}]$ represent the weight vector that connects the k^{th} output and hidden nodes, and b_k is the biases of the k^{th} hidden nodes. A more solid representation is provided as follows:

$$Y = w^{oi}x + w^{oh}G = [w^{oi}w^{oh}]\begin{bmatrix}X\\G\end{bmatrix} = W\begin{bmatrix}X\\G\end{bmatrix} \ldots (2)$$

Where

$$G\,(W_1^{in}, \cdots, W_m^{in}, W_1, \cdots, b_m, \cdots, X_N) \ldots\ldots\ldots\ldots\ldots \quad (3)$$

$$= \begin{bmatrix} g(W_1^{in}X_1 + b_1) & \cdots & g(W_1^{in}X_N + b_1) \\ \vdots & \ddots & \vdots \\ g(W_m^{in}X_1 + b_m) & \cdots & g(W_m^{in}X_N + b_m \end{bmatrix}_{m\times N}$$

$$W = [W^{oi}W^{oh}]_{I\times(n+m)} \qquad (4)$$

The matrix $W=[W^{oi}W^{oh}]$ represent the output weights while G represents the output matrix of the FLNs' hidden layer. A Moore-Penrose generalized inverse is used to resolve the model [28]. The minimum norm least-squares solution of the linear system could be expressed thus:

$$\widehat{w} = (Y)\begin{bmatrix}X\\G\end{bmatrix}^{+} \qquad (5)$$

$$w^{oi} = \widehat{w}(1{:}I, 1{:}n)$$
$$w^{oh} = \widehat{w}(1{:}I, n+1{:}n+m) \qquad (6)$$

Figure 1 presents the algorithm that explained the FLNs' learning process. This algorithm is initiated by randomly initializing the weights between the input layer and the hidden layer before proceeding to the finding of the G matrix based on the input-hidden matrix. This matrix is a representation of the hidden layers' output matrix. Next, the Moore-Penrose equations are used to find the input-output matrix (w^{oi} and w^{oh}).

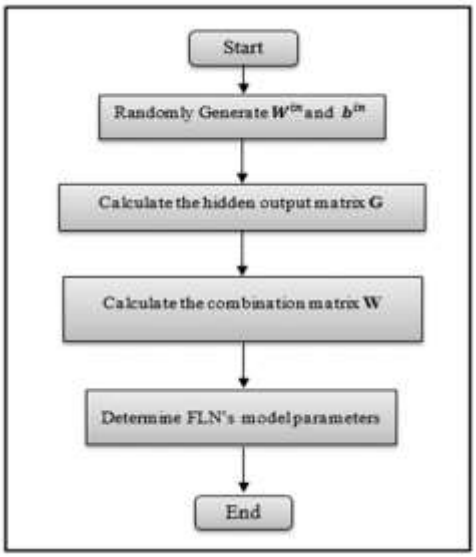

Figure.1 FLN Algorithm

4.2 Overview of Firefly Algorithm (FA)

The FA was developed as a metaheuristic framework based on inspiration from the social lifestyle of fireflies when they are in a group. Principally, each firefly randomly explores and searches for both preys and other fireflies within its vicinity. As per [29], the luminous intensity of each firefly depends on its own body-flashing pattern. The attractiveness of each firefly to the other fireflies always depends on the intensity of the light it produces (fireflies tend to be attracted to the brighter ones). Each firefly's brightness is dependent on the landscape of the objective function; hence, the differences in the intensity of light produced by individual firefly is related to the associated objective function. Therefore, the original FA was developed based on the following concepts [29]:

- It is believed that all fireflies are unisex and can be attracted to each other irrespective of sex.
- The luminous intensity of each firefly determines its level of attractiveness to the other fireflies (the brighter firefly will attract the other ones to itself). In the absence of any brighter firefly, the swarm will move randomly within the solution space.
- Each firefly's brightness depends on the landscape of its objective function.

According to [29], the brightness of each firefly is a function of the intensity of the light it produces. The differences in the intensity of the produced light are determined by the set objective function (OF). Therefore, when searching for the solution to an optimization problem, the intensity of light at location x could be proportionate to the OF $f\ (x)$ and could be determined as $I\ (x) \propto f(x)$. The light intensity $I\ (r)$ for any distance r varie,,s exponentially as:

$$I = I_0 e^{-\gamma r} \quad (7)$$

Where I_0 represent the coefficient of the original light intensity at $r = 0$ while γ represent the pre-set light absorption coefficient. The value of this relation signifies the level of attractiveness of one firefly to the others as evidenced by the strength of its light intensity. With the proportionality between the firefly's attractiveness and the produced light intensity, r could represent the distance between any 2 fireflies, while the difference in attractiveness, β could be given as:

$$\beta = \beta_0 e^{-\gamma r} \quad (8)$$

Where β_0 is the coefficient of attractiveness at $= 0$. According to Yang, the movement of firefly i towards firefly j due to the attractiveness of firefly j is determined by:

$$x_{i+1} = x_i + \beta_0 exp^{-\gamma r^2}\left(x_j - x_i\right) + \alpha\epsilon_i \quad (9)$$

Where the 3rd term represents the randomization term. This term consists of the randomization coefficient, α with the random variable vector, ϵ_i from Gaussian distribution. The following suggestions have been made for most practical applications [29]:

- Between any 2 fireflies (i and j), the distance between them at $x(i)$ and $x(j)$ is expressed by the Cartesian distance $r_{ij}^2 = (x_i - x_j)^2$.
- The randomization coefficient α is replaced by αS_k, where $\alpha\ \epsilon\ [0,1]$ and the scaling parameter S_k(k = 1,…,d) lies in the d dimensions of the real solution space of the optimisation problem.
- Both β and $\gamma\ \epsilon[0,\infty]$ are determined by the light absorption coefficient, γ. However, γ and β_0 are suggested to be =1 in practice.
- A firefly population size of n = 15 to 100 has been suggested but the actual range for practical purpose is $n = 15$ to 40.

The randomization parameter of the FA is for exploration task and proper tuning of this parameter will improve the algorithmic performance due to the trade-off established between the search for local and global optima. Contrarily, the FA uses the attractiveness parameter for the exploitation of local optimum solution especially when the optimality is near. Note that this optimal may or may not be the global optimal.

V. PROPOSED MAIN MODEL METHODOLOGY

As mentioned in the previous sections, FLN consists of three layers (input, hidden and output). These layers are connected using weights and biases. In the standard FLN, both weights and biases are generated randomly, which may affect the performance of the classification process. Therefore, generating the best values for them is an issue. In this section, the firefly algorithm (FA) is used for finding better values for both FLN parameters (weights and biases). The proposed algorithm called FA-FLN, which consists of six stages, is shown in Figure. 2 while the learning process for FA-FLN is summarized as follows:

Step1: Input

This stage is divided into three parts: FA parameters, FLN parameters, and dataset. In the first part, the main parameters of the FA algorithm are defined, including γ which is an algorithmic parameter for determining the level of dependence of the updating process on the distance between 2 two fireflies; α is the parameter that determines the step length of the randomized movement; ε () is a uniformly distributed random vector with values ranging from 0 to 1. In the second part, the number of neurons in the hidden layer (m) is defined. The third part is the dataset used.

Step2: Initialization

Each firefly in FA represents a solution, which consists of two parts, weights, and biases. The total number of variables is equal to:

$$\text{No.Vers} = \text{m} \times 2 \quad (10)$$

Where m is representing the number of neurons in the hidden layer. The number (2) represents the main parameters of basic (FLN) ($\text{W}_\text{m}^\text{in}, \text{b}_\text{m}$) equal to neurons. Each variable (position) in the firefly is initialized using the following:

$$\text{First part: } X_i^w = (U_w - L_w) \times Rand + L_w \quad (11)$$

$$\text{Second part: } X_\text{i}^\text{b} = (\text{U}_\text{b} - \text{L}_\text{b}) \times \text{Rand} + \text{L}_\text{b} \quad (12)$$

Where X_i^w represents input weight, X_i^b input basis. U_w , U_b in the equations represents the upper boundaries, L_b , $L_w$represents the lower boundaries. Rand represents a

uniformly distributed random number in the range of 0 and 1.

Step3: Fitness Function

In this stage, all the particles are evaluated using the fitness equation:

$$f(x) = 1 - A \quad (13)$$

Where $f(x)$ represent the error rate of the classification process, thus, finding a lower error rate is the main aim of FA-FLN. Therefore, this is a minimizing problem. And (A) represent the correctly classification (accuracy) sample by using FLN which is given in 14.

$$A = \frac{The\ correct\ classification}{N} \quad (14)$$

Step4: Position Update

In this step, each firefly updates its position. The new positions can be calculated using Equation. After updating, the position is determined using the following relations:

$$I = I_0 e^{-\gamma r}$$

$$\beta = \beta_0 e^{-\gamma r}$$

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(\varepsilon() - 0.5)$$

After updating the positions of the fireflies, calculate the fitness value based on the new position and compare the current best (step t).

Step5: Check Boundaries

The positions of each firefly should be checked for exceeding the upper or lower boundaries. Therefore, they should stay inside the search space of boundaries.

$$x_i = \begin{cases} U_b & , x_i > U_b \\ L_b & , x_i < L_b \end{cases} \quad (15)$$

Where U_b represent the upper boundaries; L_b represent the lower boundaries.

Step6: Termination Condition

For each iteration, the global best solution is determined. If the number of iterations has reached the maximum, then, stop the searching process and return.

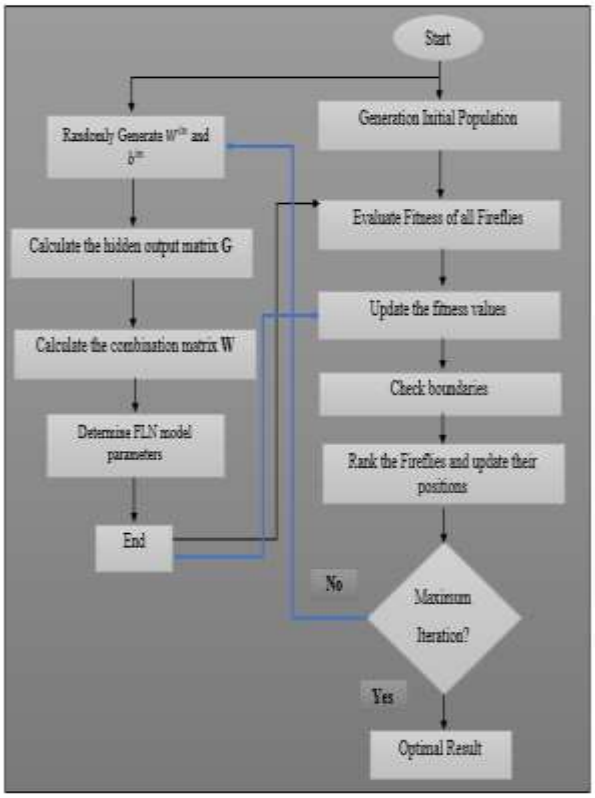

Figure 2. FA-FLN diagram

VI. OVERVIEW OF INTRUSION DETECTION SYSTEM BASED ON HYBRID MODELS & RESULTS

There are several ML frameworks that are based on IDS. [30] proposed that the current IDS research can be classified into two major domains- anomaly detection and information reduction methods. These methods mainly focus on the learning methods for alert decision support in anomaly-based ID. The FLN has been earlier demonstrated to perform better than ELM and SVM in terms of training speed, user-friendliness, and accuracy. It has been shown that ML-based ID can use FLN to extend their applicability to significantly larger datasets compared to most of the currently used datasets in most studies. This can be achieved without necessarily increasing the training time due to the near linear scaling ability of the proposed FLN.

[31] proposed a survey on the ANNs based on IDS and classified the works into simple ANN and hybrid ANN. In the simple approach, they discussed the use of BPNN, SVM, SA, and SOM for anomaly detection. The hybrid approach focused on the use of more than one technique. [32] conducted a review of the potential techniques that are based on IDS. The study covered NN, SVM, and suggested that ELMs are useful for IDS owing to their ease of implementation, fast learning speed, high generalization ability, and working with non-linear kernels and activation functions. Although other studies have suggested the usefulness of ELMs in overcoming most of the discussed challenges [33], details of previous studies on ELMs with IDS were not provided. Furthermore, there was no discussion on how to apply ELM on ID problems. They also suggested the chances of overcoming the challenges of the individual algorithms by combining different learning approaches.

[34] proposed an SVM-based filtering algorithm for the selection of multiple ID classification tasks on the NSL-KDD ID dataset. The proposed algorithm achieved 91% classification accuracy when using only 3 input features and 99% using 36 input features, while all the 41 input features of the NSL-KDD set achieved 99% classification accuracy. Meanwhile, the test set performed badly with 0.77. With this level of poor generalization efficiency, this method cannot effectively detect unknown network attacks. [35] achieved good results with Kernel-based ELM. The kernel selection is a critical step for achieving a good learning performance but the kernel-based ELM usually computes a kernel over the entire input samples and requires much memory. The computation of large datasets of a full kernel is sometimes not feasible as a result of memory problems, and in the smaller datasets that executes full kernel computation, there is a need to have a way of combining multiple classifiers or kernels to achieve good results.

[15] explored the feasibility of combining the learning decisions of multi-classifiers for the formulation of a single decision with more accuracy compared to the individual classifiers. This combination of classifiers is motivated by the fact that previous studies have demonstrated a varied classification ability of most classifiers in the detect of

specific classes in a multiclass learning problem. The introduction of a novel Multiple Adaptive Reduced Kernel ELM (MARK-ELM)-based IDS made MARK-ELM suitable for the processing of multi-class network IDS. Several techniques have been successful in the detection of several classes of attack, but their performances are often poor due to their dependence on KDD '99. The proposed approach achieved a high rate of false positives and a good detection performance which are huge challenges facing network operators.

[36] pinpointed large data volumes, low detection rate, and high false alarms as the common challenges of IDS. They used an online based sequential ELM to design an IDS-based anomaly for network traffic analysis. The performance of the proposed method was evaluated on the standard Kyoto university benchmark dataset. The feature that was used in this work was extracted from the KDD data set. The algorithm was not validated on large data sets such as KDD, hence, further validation should be performed.

A heuristic is a way of learning, discovery or problem solving which employs a practical approach that is not guaranteed to be optimal.[16] presented a GA and SVM-based anomaly detection technique. They used GA and SVM for improving the classification performance SVM. The proposed technique was evaluated on the KDDCUP '99 set. As mentioned in the limitations of SVM, it provides a binary classification as normal data or attack. Additionally, the system was only evaluated on the KDD '99 data set. Table 2 shows some of the related works based on IDS. [37] proposed an Ant Colony Optimization (ACO)-based KNN intrusion detection method. The algorithm was pre-trained with KDD Cup '99 dataset using ACO, while the performance of the KNN-ACO, BP and SVM were compared based on common performance parameters such as accuracy and false alarm rate. The study reported an overall accuracy of 94.17% and an overall FAR of 5.82% for the proposed algorithm. However, this algorithm was trained with only 26,167 samples which are relatively a small data volume.

Table.2 Related IDS works based on hybrid models

Authors	Model Type	Single	Hybrid	Algorithm	Data set	Limitations
[22]	Anomaly	-	-	PSO-Kernel FLN	10% KDD99	-The results of the proposed model didn't show the accuracy of each class, main accuracy not that accurate as the main dataset unbalance.
[14] -	Anomaly	-	-	PSO-FLN	10% KDD99	-Select randomly 10% form all the dataset. -Divided Dataset into 50% for both training and testing. Which it's not that accurate based on related work.
[38]	Anomaly	-	-	PSO-SVM	10% KDD99	-the model essay leads to a higher false alarm rate. -The model evaluates based KDD99 with all limitations
[34]	Signature	-	-	SVM	NSL-KDD	-High rate of false alarm -The performance is worse during the test set - It cannot effectively detect unknown network intrusions. -
[39]	Anomaly	-	-	Bees algorithm (BA)+ SVM	KDD cup 99	-ELM lower computational requirements than SVMs, -ELMs have shorter training time requirements than SVMs, -ELMs work directly on multi-class classification problems
[40]	Anomaly	-	-	BP + DBSCAN algorithm+	KDD cup99	-The computational cost using ELM is very small in comparison to back propagation, -Another problem of the conventional back propagation clearing algorithms is slow coverage rate
[41]	Anomaly	-	-	GA+ Decision Tree algorithm	KDD cup99	To precisely model, all the behaviors are difficult since the anomaly-basedsystems can only detect known attacks.
- [42]	- Anomaly	A -	-	Naïve Bayes , Decision Tree	NSL-KDD	Bayes needs large data sets to work, because of the assumed independence of the classes; it is also tedious to estimate the real network traffic probabilities.
[15]	Anomaly	-	-	Multiple Kernel-ELM	KDD cup99	- the author during testing mode didn't depend on the data set the testing mode to evaluate the results - This work evaluated based on KDD99, and we mentioned already the problems with this data set.
[43]	Anomaly	-	-	ELM	KDD cup99	-This work used normal ELM with the random select problem. -This work evaluated based on KDD99, and we mentioned already the problems with this data set

Table 2 showed that hybrid models achieved best accuracies compared with models based on single algorithms as mentioned in the previous section. Moreover, anomaly IDS achieved better results compared with IDS signature. On the other hand, IDS dataset represents one of the main limitations, and for models, most of the hybrid between machine learning and optimization algorithm reduced the impact of randomness when selecting the main parameter values.

VII. CONCLUSION

Intrusion detection system based on hybrid models achieved better results compared with a model based on single algorithms. However, most of these hybrid models still face several limitations which represent as motivation for proposing a new hybrid model. In addition, based on the analysis-related works presented in this work, we propose a new hybrid model called FA-FLN, consisting of the firefly algorithm and fast learning network which can overcome most of the limitations of the previous frameworks.

VIII. ACKNOWLEDGMENTS

We appreciate the financial support from the Universiti Malaysia Pahang under grant numbers RDU180344 and RDU1703287.

REFERENCE

1. 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.
2. J. M. Fossaceca, "Application of a Novel Multiple Kernel Learning Framework for Improving the Robustness of Network Intrusion Detection," no. December 1992, 2015.
3. 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.
4. E. Vasilomanolakis, S. Karuppayah, M. A. X. M. Uhlh, and M. Fischer, "55 Taxonomy and Survey of Collaborative Intrusion Detection ¨ ¨," vol. 47, no. 4, pp. 1–33, 2015.
5. M. H. Ali, M. Fadlizolkipi, A. Firdaus, and N. Z. Khidzir, "A hybrid Particle swarm optimization -Extreme Learning Machine approach for Intrusion Detection System," *2018 IEEE Student Conf. Res. Dev.*, pp. 1–4, 2019.
6. S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Futur. Gener. Comput. Syst.*, vol. 80, pp. 157–170, 2018.
7. W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, 2017.
8. M. H. Ali, K. Moorthy, M. Morad, and M. A. Mohammed, "Propose a New Machine Learning Algorithm based on Cancer Diagnosis," no. October 2018, 2019.
9. 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.
10. M. H. Ali1, "TOWARDS A EXCEPTIONAL DISTRIBUTED DATABASE MODEL FOR MULTI DBMS." pp. 553–560, 2014.
11. Bhavya Daya, "Network security: History, importance, and future," *Univ. Florida Dep. Electr. …*, p. 13, 2013.
12. T. Kaur, V. Malhotra, and D. Singh, "Comparison of network security tools-Firewall, Intrusion Detection System and Honeypot," *Int. J. Enhanc. Res. Sci. Technol. Eng.*, vol. 3, no. 2, pp. 200–204, 2014.
13. U. Kumar, "A Survey on Intrusion Detection Systems for Cloud Computing Environment," vol. 109, no. 1, pp. 6–15, 2015.
14. 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.
15. J. M. Fossaceca, T. A. Mazzuchi, and S. Sarkani, "Expert Systems with Applications 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.
16. B. M. A. R. R. M. Chizari and A. M. M. Eslami, "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1669–1676, 2016.
17. M. H. Ali, "Intrusion Detection System Framework Based on Machine Learning for Cloud Computing," no. September 2017, 2016.
18. 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.
19. M. H. Ali and M. Mohammed, "Optimize Machine Learning Based Intrusion Detection for Cloud Computing : Review Paper," no. October 2016.
20. M. H. Ali and M. A. Mohammed, "An Improved Fast Learning Network with Harmony Search based on Intrusion-Detection System," *J. Comput. Theor. Nanosci.*, vol. 16, pp. 2166–2171, 2019.
21. P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion Detection Techniques in Cloud Environment: A Survey," *J. Netw. Comput. Appl.*, vol. 77, no. October 2016, pp. 18–47, 2016.
22. M. H. Ali and M. F. Zolkipli, "Model of Improved a Kernel Fast Learning Network Based on Intrusion Detection System Model of Improved a Kernel Fast Learning Network Based on Intrusion Detection System," no. January 2019.
23. M. Zamani and M. Movahedi, "Machine Learning Techniques for Intrusion Detection," pp. 1–11, 2013.
24. X. W. Udaya Sampath K. Perera Miriya Thanthrige, Jagath Samarabandu, "Machine Learning Techniques for Intrusion Detection," *IEEE Can. Conf. Electr. Comput. Eng.*, pp. 1–10, 2016.
25. G. Li, P. Niu, X. Duan, and X. Zhang, "Fast learning network: A novel artificial neural network with a fast learning speed," *Neural Comput. Appl.*, vol. 24, no. 7–8, pp. 1683–1695, 2014.
26. Jeff Heaton, *Introduction to Neural Networks with Java*, vol. 99. 2008.
27. M. F. Z. Mohammed Hasan Ali, "Intrusion-Detection System Based on Fast Learning Network in Cloud Computing," no. September 2018.

28. N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks," *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
29. X.-S. Yang, "Firefly Algorithm, Stochastic Test Functions, and Design Optimisation," pp. 1–12, 2010.
30. J. Xiang, M. Westerlund, D. Sovilj, and G. Pulkkis, "Using Extreme Learning Machine for Intrusion Detection in a Big Data Environment," *AISec'14*, pp. 73–82, 2014.
31. B. Shah and B. H. Trivedi, "Artificial Neural Network-based Intrusion Detection System : A Survey," vol. 39, no. 6, pp. 13–18, 2012.
32. V. Jaiganesh, S. Mangayarkarasi, and P. Sumathi, "Intrusion Detection Systems : A Survey and Analysis of Classification Techniques," vol. 2, no. 4, pp. 1629–1635, 2013.
33. A. Patel, U. Universities, M. Taghavi, and K. Bakhtiyari, "An Intrusion Detection And Prevention System In Cloud Computing : A AN INTRUSION DETECTION AND PREVENTION SYSTEM IN CLOUD COMPUTING : A SYSTEMATIC REVIEW," no. December 2017, 2012.
34. M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 datasets employing SVMs," *Ski. 2014 - 8th Int. Conf. Software, Knowledge, Inf. Manag. Appl.*, 2014.
35. V. Jaiganesh and P. Sumathi, "Kernelized Extreme Learning Machine with Levenberg-Marquardt Learning Approach towards Intrusion Detection," *Int. J. Comput. Appl.*, vol. 54, no. 14, pp. 38–44, 2012.
36. R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8609–8624, 2015.
37. S. Vishwakarma, "An Intrusion Detection System using KNN-ACO Algorithm," vol. 171, no. 10, pp. 18–23, 2017.
38. H. Saxena MTech Scholar and V. Richaariya, "Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain," *Int. J. Comput. Appl.*, vol. 98, no. 6, pp. 975–8887, 2014.
39. O. Alomari and Z. A. Othman, "Bees Algorithm for feature selection in Network Anomaly detection β-Hill climbing for optimization problems View project Feature selection on high-dimensional data View project," *Artic. J. Appl. Sci. Res.*, vol. 8, no. 3, pp. 1748–1756, 2012.
40. R. Shivhare, S. Chaturvedi, and S. M. Tech, "A Novel and Hybrid Technique for Efficient Intrusion Classification," vol. 3, no. 11, pp. 9124–9127, 2014.
41. B. Senthilnayaki, K. Venkatalakshmi, and A. Kannan, "An intelligent intrusion detection system using genetic-based feature selection and Modified J48 decision tree classifier," *2013 5th Int. Conf. Adv. Comput. ICoAC 2013*, pp. 1–7, 2014.
42. D. H. Deshmukh, T. Ghorpade, and P. Padiya, "Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset," *Proc. - 2015 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2015*, 2015.
43. G.-B. Huang, "Extreme learning machines for intrusion detection," *2012 Int. Jt. Conf. Neural Networks*, pp. 1–8, 2012.