# Intrusion Detection System in IoT Based on GA-ELM Hybrid Method

Elijah M. Maseno [1,*], Zenghui Wang [2], and Fangzhou Liu [3]

[1] Department of Computer Science, University of South Africa, Florida, South Africa
[2] Department of Electrical Engineering, University of South Africa, Florida, South Africa
[3] Research Center of Intelligent Control and Systems, Harbin Institute of Technology, Harbin, China
*Correspondence: 13090879@mylife.unisa.ac.za (E.M.M)

*Abstract*—**In recent years, we have witnessed rapid growth in the application of IoT globally. IoT has found its applications in governmental and non-governmental institutions. The integration of a large number of electronic devices exposes IoT technologies to various forms of cyber-attacks. Cybercriminals have shifted their focus to the IoT as it provides a broad network intrusion surface area. To better protect IoT devices, we need intelligent intrusion detection systems. This work proposes a hybrid detection system based on Genetic Algorithm (GA) and Extreme Learning Method (ELM). The main limitation of ELM is that the initial parameters (weights and biases) are chosen randomly affecting the algorithm's performance. To overcome this challenge, GA is used for the selection of the input weights. In addition, the choice of activation function is key for the optimal performance of a model. In this work, we have used different activation functions to demonstrate the importance of activation functions in the construction of GA-ELM. The proposed model was evaluated using the TON_IoT network data set. This data set is an up-to-date heterogeneous data set that captures the sophisticated cyber threats in the IoT environment. The results show that the GA-ELM model has a high accuracy compared to single ELM. In addition, Relu outperformed other activation functions, and this can be attributed to the fact that it is known to have fast learning capabilities and solves the challenge of vanishing gradient witnessed in the sigmoid activation function.**

*Keywords*—**intrusion detection system, extreme learning machine, genetic algorithm, TON_IoT data sets, hybrid**

## I. Introduction

IoT has been adopted in different fields, which include but are not limited to smart homes, health care, farming, power grids, and smart wearables. These devices use different protocols to connect to the internet, which increases the complexity of the internet. The integration of a large number of electronic devices exposes IoT technologies to various forms of cyber-attacks [1, 2]. Cybercriminals have shifted their focus to IoT as it provides a broad network intrusion surface area. These Cyber-attacks are targeted toward breaching sensitive information, disrupting services, and breaching financial sector records. Cybercriminals have invested heavily to improve the efficiency and effectiveness of their tools. The focus of these tools is on how to evade the existing cyber defense mechanisms undetected. To achieve this, cybercriminals have sorted to invest in machine learning algorithms to develop their arsenals. Cybercriminals are leveraging ML algorithms to develop complex and intelligent cyber-attacks. These tools have the capability of learning their environment and evolving when need be. One of the standard techniques in cyber defense is Intrusion Detection Systems (IDS). IDS is used as a tool for the cyber defense to monitor and secure networks. Most of the existing IDS have proven to be ineffective in protecting IoT devices [3]. To improve the performance of the existing IDS, researchers have proposed the integration of two or more machine learning algorithms [4]. The proposed hybrid intrusion detection systems have proven to be superior to stand alone traditional intrusion detection systems. This work proposes the optimization of ELM using GA. Research shows that a major limitation of ELM is that it randomly selects the input weights and the biases. One of the commonly proposed techniques for overcoming this limitation is the use of metaheuristic algorithms [5]. Huang and Jiang *et al*. [6], Alexandre and Cuadra *et al*. [7], Matias and Araújo *et al*. [8] proposed the application of a Genetically Optimized Extreme Learning Machine (GA-ELM) in different fields. In these studies, GA was used for the optimization of input parameters or optimization of ELM structure. It was observed that the researchers proposed the use of the sigmoid activation function. The Sigmoid activation function is known to be slow in learning and in addition it suffers from vanishing gradient problem. Ali and Zolkipli *et al*. [9], in most of the existing works, the choice of activation function is always not justified but randomly selected. The study further proposes the investigation of GA-ELM using the different activation functions. GA and ELM have been widely used in the field of intrusion detection separately. Researchers have observed that most of the available data sets used for the validation of intrusion detection systems have limitations and cannot be reliably used to evaluate modern intrusion detection systems [10, 11]. Moustafa [10] observed that these data sets do not capture the complex cyber threats of

Manuscript received September 25, 2022; revised November 12, 2022; accepted January 4, 2022; published July 5, 2023.

the current digital environment and most of them are homogeneous in nature. In addition, the researcher noted that the data sets could not be used to validate modern AI-based cybersecurity solutions, due to the fact that most of the data sets were customized to validate a specific security solution. TON_IoT network data set, which is publicly available in [12], was created to solve the above mentioned limitations. TON_IoT network data set, is a new form of data set which is heterogeneous in nature and captures the complex cyber threats in the IoT environment. This data set can be used to efficiently and effectively validate AI based security solutions. To the best of our knowledge, when this research was being done no research had been done on the application of GA-ELM in the field of IDS and especially using IoT data sets. The aim of this study is first to optimize the ELM input weights using GA. Second to investigate the performance of GA-ELM using the different activation functions (sigmoid function, relu function, and sin function). The aim will be to establish whether the choice of activation function matters in the development and performance of intrusion detection systems. Third, to evaluate the performance of the model using the new generation of IoT data sets. The paper is organized as follows: Section II gives an overview of ELM, Section III gives an overview of GA, Section IV represents the proposed model, Section V represents the conducted experiments and the results and lastly section 6 provides a conclusion.

## II. Extreme Learning Machine

ELM was first proposed in 2004 by Huang *et al*. [13]. The aim of ELM was to overcome the inherent limitation of the classical feed-forward neural networks. The major limitation of the feed-forward neural networks is that they are slow due to the use of gradient-based learning algorithms for training. With gradient descent learning algorithms the parameters, i.e., weights and biases in one layer depend on parameters on other layers and are prone to converge to local minima [13, 14]. Over the years researchers have proposed solutions to improve the performance of feed-forward neural networks. As part of the solutions, ELM has achieved great success in improving the performance of the single hidden layer feedforward neural networks (SLFNs). ELM randomly initializes the parameters that connect the input layer with the hidden layer, the output weights are obtained using the least-square technique [5, 13, 14]. The fast learning capabilities of ELM can be attributed to the fact that it learns without iteration, which makes ELM to converge much faster compared to the classical feed-forward neural networks. Randomization of the input parameters in ELM eliminates the problem of local minima found in the classical networks. ELM outperforms other learning algorithms in terms of learning speed, ease of implementation, and generalization performance. [15]. Due to these attributes, ELM has found its application in different fields, which include but are not limited to regression, classification, and clustering.

Given N distinct training set $(\mathbf{X}_i, \mathbf{t}_i)$, where $\mathbf{X}_i = [x_{i_1}, x_{i_2}, \ldots, x_{i_n}]^T \in \mathbf{R}^n and \mathbf{t}_i = [t_{i_1}, t_{i_2}, \ldots, t_{i_m}]^T \in \mathbf{R}^m$. L and $g(x)$ represents a number of hidden nodes and activation function respectively. ELM can be implemented by randomly assigning the parameters of the hidden nodes$(\omega, b)$, computing the hidden layer output matrix $(\mathbf{H})$ and the output weights $(\boldsymbol{\beta})$. Using N samples, our target output $\mathbf{T}$ can be obtained using the equation below:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{1}$$

where

$$\mathbf{H}(\omega_1, \cdots, \omega_L, \quad b_1, \cdots, b_L, \quad x_1, \cdots, x_N) \tag{2}$$

$$\begin{bmatrix} g(\omega_1, x_1 + b_1) & , \cdots , & g(\omega_L, x_1 + b_L) \\ \vdots & , \cdots , & \vdots \\ g(\omega_1, x_N + b_1) & , \cdots , & g(\omega_L, x_N, b_N) \end{bmatrix}_{NL} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{Lm} \tag{4}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{Nm} \tag{5}$$

To compute the weights connecting the hidden layer and the output layer represented by $\boldsymbol{\beta}$, the least-squares technique is applied to minimize the error between the target and the output [13, 15].

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$is the Moore–Penrose generalized inverse of matrix **H,** and **T** is the target [16].

## III. Genetic Algorithm

Genetic algorithm is a type of Evolution Algorithm which operate on the principles of evolution and natural selection [17]. In GA a set of chromosomes are randomly selected to represent the problem to be solved. These sets of chromosomes are referred to as population during the phases of evolution. An evaluation function is used to find the best candidate for each chromosome. The mutation of the species is guided by crossover and mutation. Reference [18], when a GA is used for problem-solving, three factors will have an impact on the effectiveness of the algorithm: 1) the selection of fitness function; 2) the representation of individuals; 3) the values of the GA parameters.

## IV. Proposed GA-ELM Optimization

To start the process of optimization, we set the number of hidden neurons and activation function. After the initial configuration of the ELM network, the initial input values are randomly generated. The input values consist of weights and biases. The model is trained in order to extract the best output values. The output values from ELM will form the initial population of GA. At this point, we will apply the principles of evolution. Genetic Algorithms are a family of bio-inspired metaheuristic optimization algorithms that leverage on the Darwinian theory of

evolution to come up with the best weights for an optimization problem. Using the survival of the fittest principle from natural selection, the core concept in identifying the most optimum weights is through the computation of a fitness score. In this work, the implementation of the GA was initiated by declaring each attribute of the intrusion except the type of the intrusion as an equation input, i, and the entire set of observations of the intrusions as the population, n. A fitness score was then computed using the formula below:

$$fitness - score = \sum_{n=1}^{n} n \times i \qquad (7)$$

While the choice of fitness score may be arbitrary, it is advisable to choose a fitness score whose computation will not be expensive as this would result in a bottleneck in the computation speed of the entire genetic algorithm.

Since natural selection favors the fittest, the objective of calculating the fitness score for each individual in the set of equation inputs was to identify individuals with the best fitness scores. Collectively, these individuals would form the mating pool that would create subsequent generations. For this work, the number of mating parents was set to 4. The choice of a number of mating parents can be justified by the fact that in some populations, increasing the number of mating partners may significantly increase the quality of the offspring produced.

Mating among the parents was achieved by defining a function that mimicked crossing in animals. In this process, a portion from each parent was identified to be used to form part of the offspring. Each parent, therefore, produced this portion which collectively made up the entire set of genes for the offspring. In this work, half of each parent's genes were used in creating the genetic structure of the offspring.

Mutation is a process that results in changes in the genetic structure of an organism, therefore differentiating one organism from another at various levels. In natural selection, mutation is critical in determining the survival or possible extinction of an organism. To simulate mutation, an integer was selected at random between −1 and 1000 in steps of 1 to represent a mutation agent. The mutating agent was incorporated into the genetic makeup of the offspring by directly adding it to a randomly selected site in the genetic structure of the individual. This new addition would then be duplicated to another random site to ensure that the mutation would occur at random locations in the entire structure of the individual. Finally, this process was let to run for 100 iterations to simulate evolution over 100 epochs.

If the termination condition is satisfied, the optimized weights are uniformly selected to retrain the model. This selection was bound by the minimum and maximum values of the weights obtained from the last generation of the GA evolution. The best solution was recorded and a comparison was thereafter made based on the accuracy of the weights obtained from genetic algorithms, and random uniform generation.

The activation functions used in this study include the sigmoid function, relu function, and sin function. The output of the hidden layer was then obtained from the dot product of the output of the input layer and the beta weights from the hidden layer.

Fig. 1 below depicts the process of GA-ELM optimization in a flow diagram.

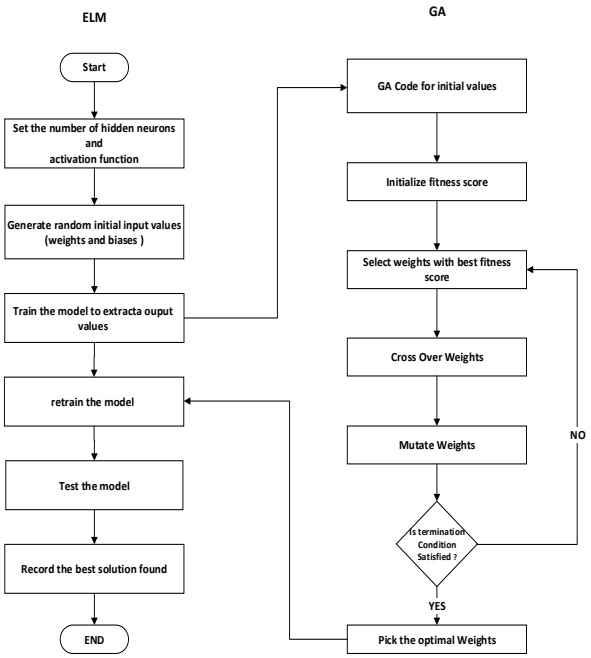


Figure 1. GA-ELM flow diagram.

## V. Experimental Results

To test the performance of the model, $TON\_IoT$ network data set was used. In this study, 80% of the data set was used for training the model, and 20% of the data set was used for testing the model. Using different activation functions the performance of GA-ELM was compared with the stand-alone ELM. The comparison of the algorithms was based on accuracy and running time. The number of hidden neurons was set to 100.

Tables I–III show the performance of ELM and GA-ELM using different activation functions. As shown in Fig. 2, ELM performed poorly in terms of accuracy with the use of the sigmoid function. On the other hand, GA-ELM accuracy performance reduced significantly with the use of the Sin activation function. ELM recorded a slightly high accuracy with the use of the Sin activation function compared to the use of the Relu activation function. GA-ELM recorded the highest results of 97% with the use of the Relu activation function. The results showed the optimization of ELM using GA increased the accuracy of ELM. The use of Relu and Sigmoid slightly improved the running time of the GA-ELM model as shown in Fig. 3. We further compared the performance of GA and GA-ELM using other metrics such as Precision, Recall, and f1-score. In this comparison, we used the Relu activation function only. The results show that GA-ELM outperformed ELM in all three metrics. As shown in Fig. 4, GA-ELM recorded 96.44% precision rate, this show that the model had a low false positive rate compared to ELM which had a precision rate of 92%. Most of the existing

IDS face the challenge of a high false positive rate, which reduces their performance. On the other hand, GA-ELM recorded a Recall rate of 98.18% which was also an improvement compared to stand-alone ELM. Finally, GA-ELM had an f1-Score rate of 97.7%, while ELM recorded f1-Score rate of 91.78%.

TABLE I. PERFORMANCE OF THE ALGORITHMS USING RELU

| Algorithm | Activation function | Running Time | Hidden Neurons | Accuracy |
|---|---|---|---|---|
| ELM | Relu | 0.382 | 100 | 91.95 |
| GA-ELM | Relu | 0.302 | 100 | 97.68 |

TABLE II. PERFORMANCE OF THE ALGORITHMS USING SIGMOID

| Algorithm | Activation function | Running Time | Hidden Neurons | Accuracy |
|---|---|---|---|---|
| ELM | Sigmoid | 0.386 | 100 | 73.04 |
| GA-ELM | Sigmoid | 0.312 | 100 | 90.33 |

TABLE III. PERFORMANCE OF THE ALGORITHMS USING SIN

| Algorithm | Activation function | Running Time | Hidden Neurons | Accuracy |
|---|---|---|---|---|
| ELM | Sin | 0.253 | 100 | 93.01 |
| GA-ELM | Sin | 0.31 | 100 | 52.50 |

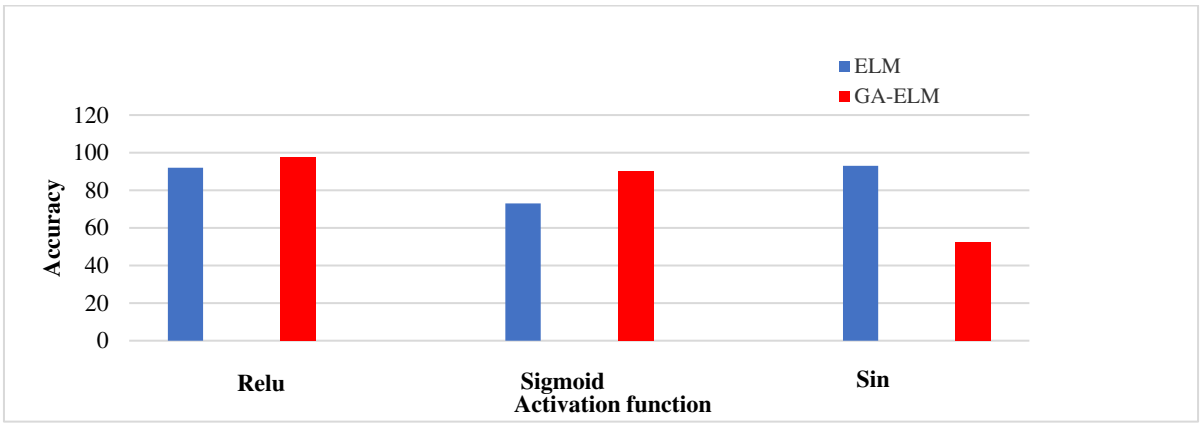


Figure 2. Accuracy comparison (GA-ELM vs ELM).

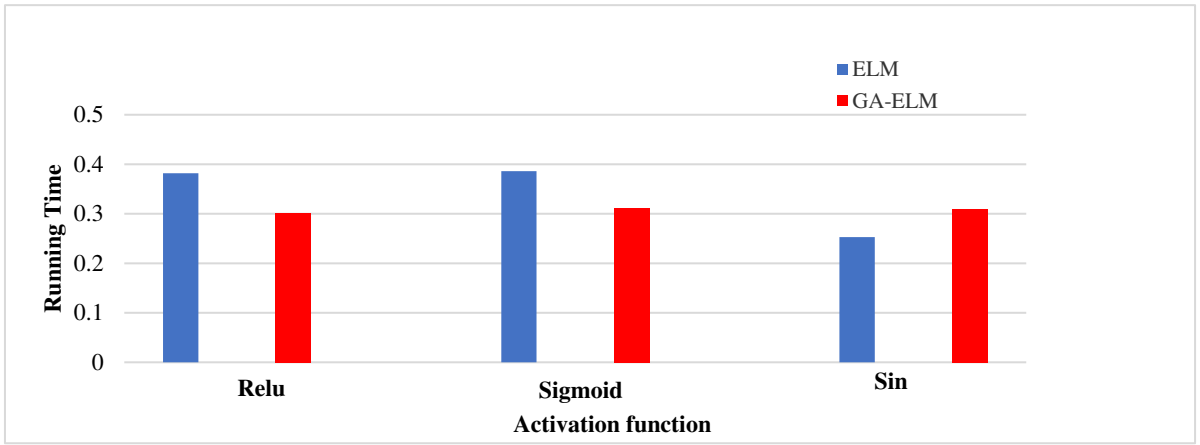


Figure 3. Running time comparison (GA-ELM vs ELM).

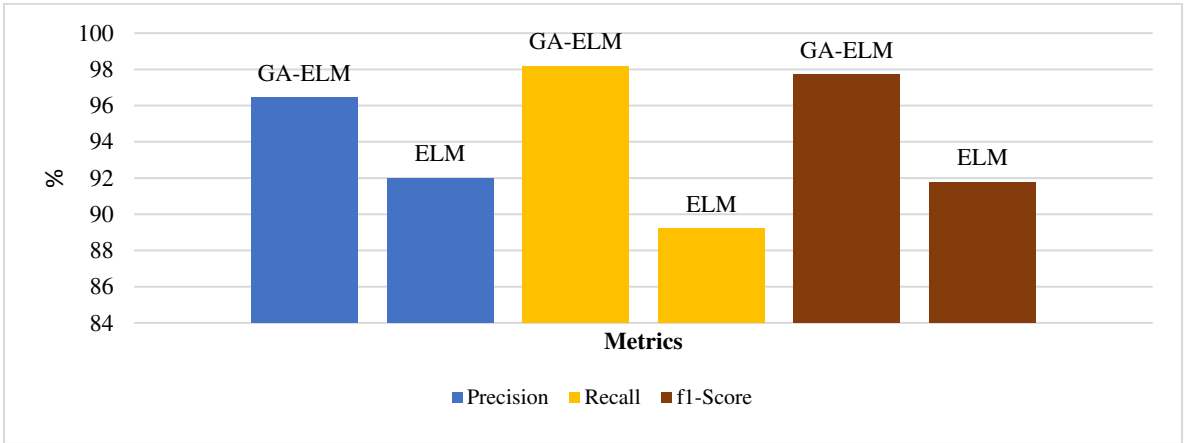


Figure 4. Precision, recall and f1-score results of GA-ELM and ELM using Relu activation function.

## VI. Conclusions

In this work, we proposed a hybrid intrusion detection system based on ELM and GA. In this study, GA was used for the selection of input weights of the ELM. To validate the choice of activation function we tested the performance of the models using different activation functions, which include: sigmoid function, relu function, and sin function. To test the models, we used TON_IoT network data set, which captures the complex cyber threats in the IoT environment. The results of the experiments show that the optimization of ELM using GA increases the performance of the model. We compared the performance of ELM against GA-ELM using different metrics, which include: accuracy, precision, recall, and f1-score. GA-ELM outperformed ELM in all these intrusion detection performance measurements. We observed that most of the researchers never disclose the choice of activation function, and one of our goals was to investigate the choice of activation function in model development. With this research, we have proven that the choice of activation function is of great importance in the performance of the models. Most of the existing works in GA-ELM have used Sigmoid activation function, we have demonstrated that this could not be the right choice, it performed poorly in our experiments compared to relu activation function. We highly recommend the use of relu, which performed better in this research. This can be investigated further with different data sets to eliminate biases if any. In this research, we focused only on the optimization of the input parameters, in the future we recommend the optimization of the ELM structure together with the input parameters for further improvement of the model.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

Elijah M. Maseno reviewed the literature, designed the research methodology, collected the results, and compiled the manuscript under the supervision of Zenghui Wang. Fangzhou Liu edited the paper and confirmed the results. All authors reviewed the results and approved the final version of the manuscript.

## Funding

This research was supported by South African National Research Foundation Grants (Nos. 114911, 137951 and 132797) and Tertiary Education Support Programme (TESP) of South African ESKOM.

## References

[1] J. Liu, D. Yang, M. Lian, and M. Li, "Research on intrusion detection based on particle swarm optimization in IoT," *IEEE Access*, vol. 9, pp. 38254–38268, 2021.

[2] N. Moustafa, M. Ahmed, and S. Ahmed, "Data analytics-enabled intrusion detection: Evaluations of ToN_IoT linux datasets," in *Proc. the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 727–735.

[3] L. Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A deep learning-based DDoS detection framework for internet of things," in *Proc. the 2020 IEEE International Conference on Communications (ICC 2020)*, 2020, pp. 1–6.

[4] U. S. Musa, S. Chakraborty, M. M. Abdullahi, and T. Maini, "A review on intrusion detection system using machine learning techniques," in *Proc. the 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 541–549.

[5] M. Eshtay, H. Faris, and N. Obeid, "Metaheuristic-based extreme learning machines: A review of design formulations and applications," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 1543–1561, 2019.

[6] Q. Huang, C. Jiang, and Y. Huang, "The prediction method of SO2 concentration in sulfuric acid production process based on GA-ELM," in *Proc. the 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC),* 2016, pp. 140–143.

[7] E. Alexandre, L. Cuadra, J. C. Nieto-Borge, G. Candil-García, M. del Pino, and S. Salcedo-Sanz, "A hybrid genetic algorithm — Extreme learning machine approach for accurate significant wave height reconstruction," *Ocean Modelling*, vol. 92, pp. 115–123, 2015.

[8] T. Matias, R. Araújo, C. H. Antunes, and D. Gabriel, "Genetically optimized extreme learning machine," in *Proc. the 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2013, pp. 1–8.

[9] 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," *Journal of Engineering and Applied Sciences*, vol. 12, pp. 4180–4185, 2017.

[10] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network ton IoT datasets," *Sustainable Cities and Society*, vol. 72, 102994, 2021.

[11] E. M. Maseno, Z. Wang, and H. Xing, "A systematic review on hybrid intrusion detection system," *Security and Communication Networks*, 9663052, 2022.

[12] TON_IoT datasets. (2020). [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, 2004, pp. 985–990.

[14] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, "Extreme learning machine: Algorithm, theory and applications," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 103–115, 2006.

[15] J. Wang, S. Lu, S. H. Wang, and Y. D. Zhang, "A review on extreme learning machine," *Multimed Tools Appl*, 2021.

[16] D. Serre, *Matrices: Theory and Applications*, SpringerLink, 2002.

[17] W. Li, "Using genetic algorithm for network intrusion detection," in *Proc. the United States Department of Energy Cyber Security Group 2004 Training Conference*, Kansas City, Kansas, 2004, pp. 24–27.

[18] R. H. Gong, M. Zulkernine, and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection," in *Proc. the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, 2005, pp. 246–253.