Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.030188
Article

Tech Science Press

Automated Machine Learning Enabled Cybersecurity Threat Detection in Internet of Things Environment

Fadwa Alrowais[1], Sami Althahabi[2], Saud S. Alotaibi[3], Abdullah Mohamed[4], Manar Ahmed Hamza[5,*] and Radwa Marzouk[6]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, 11671, Saudi Arabia
[2]Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[3]Department of Information Systems, College of Computing and Information System, Umm Al-Qura University, Saudi Arabia
[4]Research Centre, Future University in Egypt, New Cairo, 11745, Egypt
[5]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam Bin Abdulaziz University, AlKharj, Saudi Arabia
[6]Department of Mathematics, Faculty of Science, Cairo University, Giza, 12613, Egypt
*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa
Received: 20 March 2022; Accepted: 20 April 2022

Abstract: Recently, Internet of Things (IoT) devices produces massive quantity of data from distinct sources that get transmitted over public networks. Cybersecurity becomes a challenging issue in the IoT environment where the existence of cyber threats needs to be resolved. The development of automated tools for cyber threat detection and classification using machine learning (ML) and artificial intelligence (AI) tools become essential to accomplish security in the IoT environment. It is needed to minimize security issues related to IoT gadgets effectively. Therefore, this article introduces a new Mayfly optimization (MFO) with regularized extreme learning machine (RELM) model, named MFO-RELM for Cybersecurity Threat Detection and classification in IoT environment. The presented MFO-RELM technique accomplishes the effectual identification of cybersecurity threats that exist in the IoT environment. For accomplishing this, the MFO-RELM model pre-processes the actual IoT data into a meaningful format. In addition, the RELM model receives the pre-processed data and carries out the classification process. In order to boost the performance of the RELM model, the MFO algorithm has been employed to it. The performance validation of the MFO-RELM model is tested using standard datasets and the results highlighted the better outcomes of the MFO-RELM model under distinct aspects.

Keywords: Cybersecurity threats; classification; internet of things; machine learning; parameter optimization

1 Introduction

Internet of Things (IoT) is an enormous arrangement of gadgets associated through the private or public web, and that is implanted with the capacity to converse with one another streaming continuous information

with less or no mediation expected from people, consequently constructing a brought together knowledge [1]. These days, gadgets of any size with a chip introduced for empowering concentrated control, gadget to-gadget control, remote sensor organizations, and installed frameworks are viewed as IoT gadgets [2]. For instance, security movement sensors, cell phones, voice associate controlled home mechanization gadgets like TVs, speakers, home lighting frameworks are viewed as IoT gadgets. The IoT empowered advancements can be utilized to foster savvy urban communities, school systems, e-shopping, e-banking, keep up with our wellbeing, oversee industry, and engage and safeguard people [3].

The IoT gadgets can be utilized for an open assault because of generally accessible to the organization. The modern IoT-cloud can be effortlessly designated by malware contamination and pilfered programming for hurtful utilization and to think twice about [4]. The product robbery is the improvement of programming by reusing source codes illicitly from another person's work and camouflage as the first form. The saltine might duplicate the rationale of the first programming by figuring out strategies and afterward plan a similar rationale in one more kind of source code. It is an extreme danger to web security, which gives admittance to limitless downloads of pilfered programming, open source codes, and, advances and publicizes of pilfered adaptations. It quickly expands every year and gives significant financial misfortune to the product business [5].

At present, malicious assaults are all the more effectively unexpected because of the developing number of IoT organizations. The malware assaults are normally wanted to taint the security of IoT hubs, PC frameworks, and cell phones over the web. The malware recognizable proof examination is isolated into two principle techniques, for example, static and dynamic methodologies. In unique methodology, malware designs are learned while executing code in an ongoing virtual climate. Malicious way of behaving can be seen by work calls, work boundaries' investigation, information stream, guidance follows, and visual examination of codes. Reports express that a large portion of the assault traffic created on IoT networks is mechanized through different means like content and malware [6]. The expansion in assaults joined with the independent idea of the chases down is an issue for IoT networks as the gadgets utilized in a fire and neglect design for a long time with practically no human connection.

The challenges involved in the IoT gadgets including restricted handling power and transmission capacity imply that giving sufficient security can be troublesome, which can bring about network layer goes after like denial of service (DoS) [7]. Consequently, it is critical to explore ways of distinguishing this sort of traffic on networks, which can be utilized in intrusion location and anticipation frameworks. Machine learning (ML) strategies can be taken advantage of to identify malicious traffic in intrusion recognition and avoidance frameworks. ML is a subset of artificial intelligence (AI) that includes utilizing calculations to gain from information and make expectations because of the information given [8]. ML has numerous applications remembering for retail, medical care, and money where AI calculations might be applied for foreseeing client ways of managing money, anticipating clinical issues in patients, and recognizing bank extortion, individually. Because of the enormous yearly expansions in cyberattacks that are being seen consistently, ML techniques are being fused to assist with handling the rising dangers of cyberattacks. ML includes a few purposes inside the field of cybersecurity, for example, network danger investigation, which can be characterized as the demonstration of examining dangers to the organization. ML can be gainful in this errand as it can screen approaching and active traffic to recognize possibly dubious traffic [9]. ML can be applied to intrusion detection systems (IDS) to assist with further developing the capacity of the framework to run independently and increment the precision of the framework while raising the alert on a speculated assault [10].

Novo et al. [11] present the study and estimation of many pre-processed approaches dependent upon traffic categorization to an ML-NN technique. The objective of this research was for evaluating this categorized by utilizing different data pre-processed approaches for obtaining one of the accurate

methods. The presented depicts that, with executing the categorization of network traffic and many pre-processed approaches, the accuracy is improved by up to 45%. In [12], a smart IDS suitable for detecting IoT based attacks was executed. Particularly, for detecting malicious IoT network traffic, a DL technique was utilized. An identity solution makes sure the security of function and supports the IoT connectivity protocol for interoperating.

In [13], a novel approach auto-encoder DNN (AENN) was established with assuming evasion, priority violation, exploratory, and causative attack. The generated approach classifications the broadcast results utilized for predicting the transmit state if it can be jam data broadcast or sensing data. Next, the sensing data was executed to network trained which forecasts the intermediate attack. The authors in [14] present a robust scheme specially for helping detect botnet attacks of IoT devices. It can be complete by innovatively integrating the method of CNN-LSTM technique progress for detecting 2 general and serious IoT attacks (BASHLITE and Mirai) on 4 kinds of security cameras. The datasets that limited normal malicious network packet is gathered in real-time lab linked camera device from IoT environment.

This article introduces a new Mayfly optimization (MFO) with regularized extreme learning machine (RELM) model, named MFO-RELM for Cybersecurity Threat Detection and classification in IoT environment. The presented MFO-RELM technique accomplishes the effectual identification of cybersecurity threats that exist in the IoT environment. For accomplishing this, the MFO-RELM model pre-processes the actual IoT data into a meaningful format. In addition, the RELM model receives the pre-processed data and carries out the classification process. In order to boost the performance of the RELM model, the MFO algorithm has been employed to it. The performance validation of the MFO-RELM model is tested using standard datasets.

2 Design of MFO-RELM Model

In this study, a new MFO-RELM model is introduced for Cybersecurity Threat Detection and classification in IoT environment. The presented MFO-RELM technique accomplishes the effectual identification of cybersecurity threats that exist in the IoT environment. At the initial stage, the MFO-RELM model pre-processes the actual IoT data into a meaningful format. In addition, the RELM model receives the pre-processed data and carries out the classification process. In order to boost the performance of the RELM model, the MFO algorithm has been employed to it. Fig. 1 shows the workflow of MFO-RELM model.

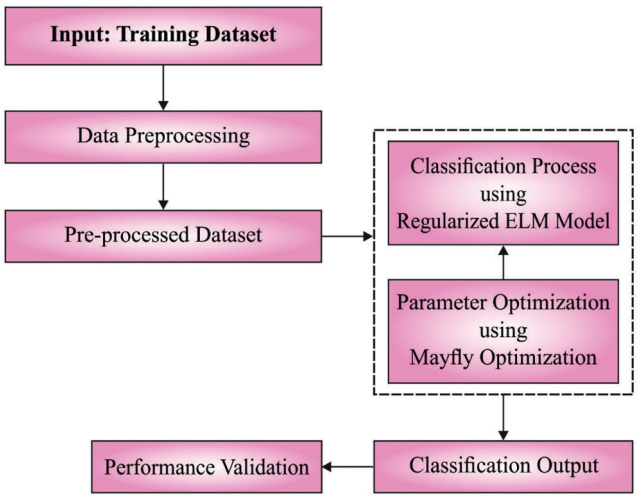

Figure 1: Workflow of MFO-RELM model

2.1 Process Involved in RELM Classification

After pre-processing, the RELM model receives the pre-processed data and carries out the classification process. The preprocessed data is given as input to the RELM technique for accomplishing classifier procedure. The SLFN techniques as BP learning approach are widely utilized ML approaches under the case from various regions [15]. This technique decreases the cost function for retaining the precision in an appropriate range by searching the particular input weighted and hidden layer (HL) bias which outcomes in improving in computation cost. An ELM is an effective solution for SLFN. The SLFN with L hidden node and activation function $g(x)$ was provided under

$$Y_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta_i, \tag{1}$$

In which $\beta = [\beta_1, \ldots, \beta_2]^T$ signifies the resultant weighted matrix amongst the output and hidden nodes. $h_i(x)$ stands for the hidden node outcome. Dissimilar from SVM and other *BP* based techniques, the parameter of HLs as the input weight w_i and HL bias b_i does not have that tuned and is arbitrarily created before the trained sample is obtained. For providing N trained instances $\{(x_j,\ t_j)\}_{j=1}^{N}$, ELM solved the learning problem by minimalized the error amongst t_j & Y_j:

$$\|H(w_1, \ldots,\ w_{\tilde{N}},\ b_1, \ldots,\ b_{\tilde{N}})\hat{\beta} - T\| = \min_{\beta} \|H\hat{\beta} - T\| \tag{2}$$

whereas

$$H(w_1,\ \ldots, w_{\tilde{N}},\ b_1,\ \ldots, b_{\tilde{N}}) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \ldots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}, \tag{3}$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}.$$

At this point, H is designated the HL resultant matrix. The resultant weighted β is calculated as:

$$\beta = H^{+}T, \tag{4}$$

In which H^{+} refers to the Moore Penrose generalizing inverse of matrix H with benefits of speed. For increasing the precision, ELM was combined as sparse demonstration. This hybrid model carries out classifiers from two fundamental phases [16]. Primarily, the ELM network was trained with the convention trained method. Conversely, during the testing step, reliability based classifier was utilized. In reliability-based classifier, the ELM classifications were executed once the tested information was classified correctly; then, the sparse demonstration based classifiers were utilized. Also, a normalized term was comprised to increase generalized performance and generate the solution further robust. At last, the resultant weighted of RELM is provided here:

$$\beta = \left(\frac{I}{C} + H^T H\right)^{-1} H^T T \tag{5}$$

2.2 Process Involved in MFO Based Parameter Optimization

To boost the performance of the RELM model, the MFO algorithm has been employed to it. The selection of variables in the RELM model is vital for accomplishing an effective classifier outcome. Mostly, the ML method involves parameters that need to be enhanced. While the trial-and-error method is not possible, metaheuristic optimization related MFO approach is exploited to select variables. Commonly, the predictive error function performs as objective function of MFO method [17]. In MF, in swarm for MO method is detached into female and male MFs. As well, the male MF is strong, it will perform as optimal in optimization. In comparison with specific swarm in PSO method, the MO method is upgrading the location based on the existing position $p_i(t)$ and velocity $v_i(t)$ at the existing round:

$$p_i(t+1) = p_i(t) + v_i(t+1) \tag{6}$$

All the female and male MFs upgrade their location. But the velocity can be upgraded in different methods. The process involved in the MFO algorithm is given in Fig. 2.

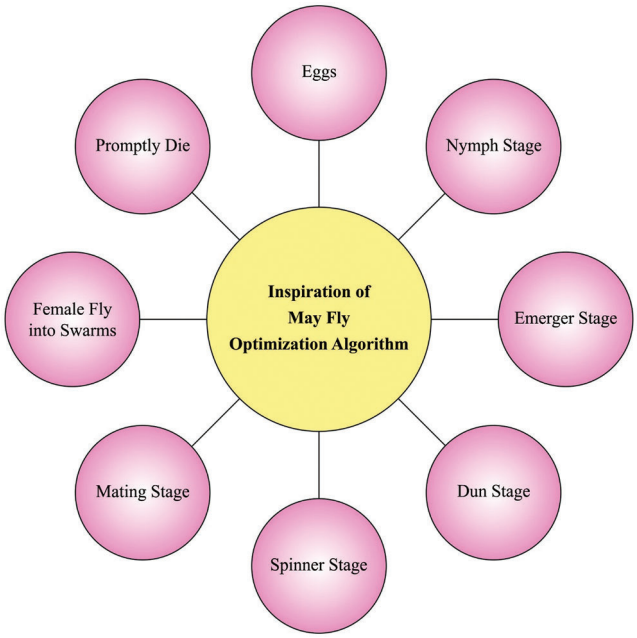

Figure 2: Process involved in MFO algorithm

2.2.1 Movements of Male MFs

They implemented exploration or exploitation procedures in iteration. The velocity is upgraded by the existing fitness values $f(x_i)$ and the previous optimum fitness value in path $f(x_{h_i})$. When $f(x_i) > f(x_{h_i})$, the male MF updates its velocity according to the existing velocity, integrated to the distance amongst others and the global optimal position, the previous finest path:

$$v_i(t+1) = g \cdot v_i(t) + a_1 e^{-\beta r_p^2} \left[x_{h_i} - x_i(t)\right] + a_2 e^{-\beta r_q^2}, \left[x_g - x_i(t)\right] \tag{7}$$

While g indicates the variable decreased linearly in the highest to lowest value. a_1, a_2, & β represent constant for balancing the value. r_p & r_g characterizes parameter employed for the Cartesian distance among the previous optimal position and the individuals, the global best position in swarm.

The Cartesian distance is the succeeding average to the distance collection:

$$||x_i - x_j|| = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{8}$$

Alternatively, when $f(x_i) < f(x_{h_i})$, the male MF update its velocity in the existing one with subjective dance coefficient d:

$$v_i(t+1) = g \cdot v_i(t) + d \cdot r_1 \tag{9}$$

Then, r_1 represents the haphazard quantity from the standard distribution and is elected within −1 & 1 [18].

2.2.2 Movements of Female MFs

They update its velocity in numerous methods. Usually, female MF with wings live around 1–7 days, henceforth the female MF is urgency to distinguish the male MF to reproduce and mate them. Therefore, update their velocity based on the male MF. In the MO technique, the topmost optimal male and female MFs are described by the earlier mating, also the subsequent finest female, male MF is described by the succeeding mates, and so on. Therefore, the i-th female MF, once $f(y_i) < f(x_i)$:

$$v_i(t+1) = g \cdot v_i(t) + a_3 e^{-\beta r_{mf}^2} [x_i(t) - y_i(t)] \tag{10}$$

Here, a_3 suggests constant and is employed for balancing the velocity. r_m indicates the Cartesian distance amongst others. Alternatively, when $(y_i) < f(x_i)$, the female MF update its velocity in the existing one with additional subjective dance fl:

$$v_i(t) = g \cdot v_i(t) + fl \cdot r_2 \tag{11}$$

While r_2 embodies the accidental quantity from a standard distribution within −1 & 1.

2.2.3 Mating of MFs

Every topmost half female and male MFs are reproduced and set pair of children to others. The offspring are established randomly as follows:

$$offspring1 = L * male + (1 - L) * female \tag{12}$$

$$offspring2 = L * female + (1 - L) * male \tag{13}$$

In which L indicates the haphazard amount from Gauss distribution.

3 Experimental Validation

This section inspects the performance validation of the MFO-RELM model on test N-BaIoT dataset [19]. It comprises samples under several class labels.

Fig. 3 illustrates the confusion matrix offered by the MFO-RELM model on entire dataset. The figure reported that the MFO-RELM model has identified 982 samples under Mirai udpplain, 994 samples under Mirai udp, 990 samples under Mirai synm, 991 samples under Mirai scan, 989 samples under Mirai ack, 979 samples under Gafgyt udp, 996 samples under Gafgyt tcp, 994 samples under Gafgyt scan, 983 samples under Gafgyt junk, 986 samples under Gafgyt combo, and 988 samples under Benign class.

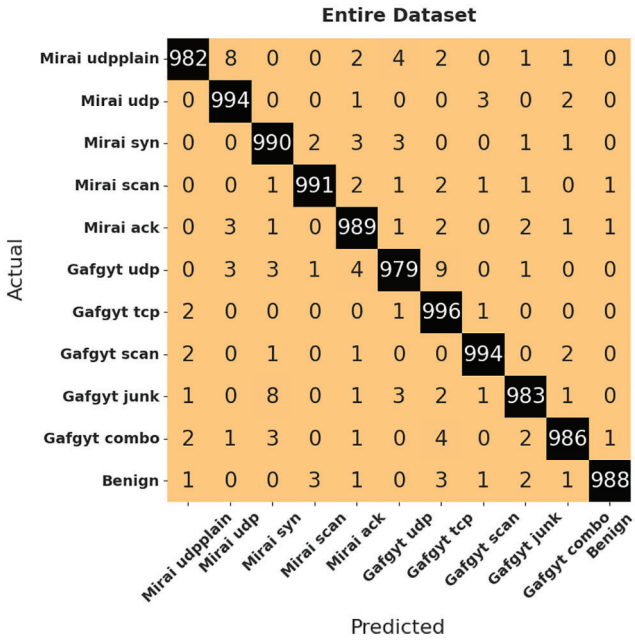

Figure 3: Confusion matrix of MFO-RELM model on entire dataset

Tab. 1 and Fig. 4 exhibits detailed classification outcomes of the MFO-RELM model on entire dataset. The results implied that the MFO-RELM model has accomplished effecula results in each class. For instance, in Mirai udpplain class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.76%, 99.19%, 98.20%, and 98.69% respectively. Along with that, in Mirai udp class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.81%, 98.51%, 99.40%, and 98.95% respectively. In addition, in Mirai syn class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.75%, 98.31%, 99%, and 98.65% respectively. Moreover, in Mirai udpplain class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.86%, 99.40%, 99.10%, and 99.25% respectively.

Table 1: Classification results of MFO-RELM model on entire set

Complete dataset				
Class labels	Accuracy	Precision	Recall	F-score
Mirai udpplain	99.76	99.19	98.20	98.69
Mirai udp	99.81	98.51	99.40	98.95
Mirai syn	99.75	98.31	99.00	98.65
Mirai scan	99.86	99.40	99.10	99.25
Mirai ack	99.75	98.41	98.90	98.65
Gafgyt udp	99.69	98.69	97.90	98.29
Gafgyt tcp	99.75	97.65	99.60	98.61
Gafgyt scan	99.88	99.30	99.40	99.35
Gafgyt junk	99.75	98.99	98.30	98.65
Gafgyt combo	99.79	99.10	98.60	98.85
Benign	99.86	99.70	98.80	99.25
Average	99.79	98.84	98.84	98.84

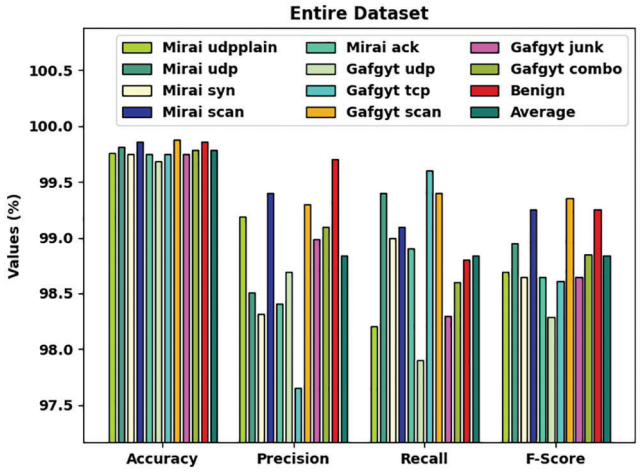

Figure 4: Classification outcomes of MFO-RELM model on entire dataset

Fig. 5 exemplifies the confusion matrix presented by the MFO-RELM model on 90% of training dataset. The figure stated that the MFO-RELM model has identified 676 samples under Mirai udpplain, 698 samples under Mirai udp, 697 samples under Mirai syn, 693 samples under Mirai scan, 719 samples under Mirai ack, 677 samples under Gafgyt udp, 684 samples under Gafgyt tcp, 710 samples under Gafgyt scan, 692 samples under Gafgyt junk, 687 samples under Gafgyt combo, and 674 samples under Benign class.

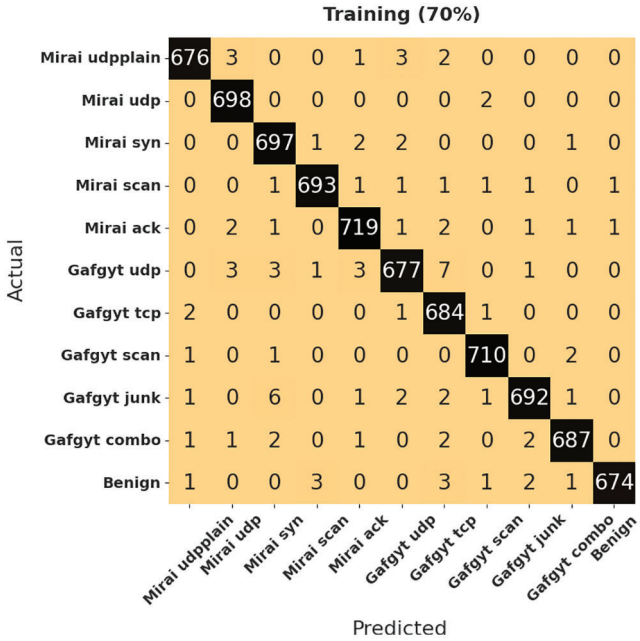

Figure 5: Confusion matrix of MFO-RELM model on 30% of training dataset

Tab. 2 and Fig. 6 display thorough classification outcomes of the MFO-RELM model on 70% of training dataset. The results implied that the MFO-RELM model has achieved effective results in each class. For instance, in Mirai udpplain class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.81%, 99.12%, 98.69%, and 98.90% respectively. Eventually, in Mirai udp class, the MFO-RELM

model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.86%, 98.73%, 99.71%, and 99.22% respectively. Meanwhile, in Mirai syn class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.74%, 98.03%, 99.15%, and 98.59% respectively. Followed by, on Mirai udpplain class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.84%, 99.28%, 99%, and 99.14% respectively.

Table 2: Classification results of MFO-RELM model on 70% of training set

Training set (70%)				
Class labels	Accuracy	Precision	Recall	F-score
Mirai udpplain	99.81	99.12	98.69	98.90
Mirai udp	99.86	98.73	99.71	99.22
Mirai syn	99.74	98.03	99.15	98.59
Mirai scan	99.84	99.28	99.00	99.14
Mirai ack	99.77	98.76	98.76	98.76
Gafgyt udp	99.64	98.54	97.41	97.97
Gafgyt tcp	99.70	97.30	99.42	98.35
Gafgyt scan	99.87	99.16	99.44	99.30
Gafgyt junk	99.73	99.00	98.02	98.51
Gafgyt combo	99.81	99.13	98.71	98.92
Benign	99.83	99.70	98.39	99.04
Average	99.78	98.80	98.79	98.79

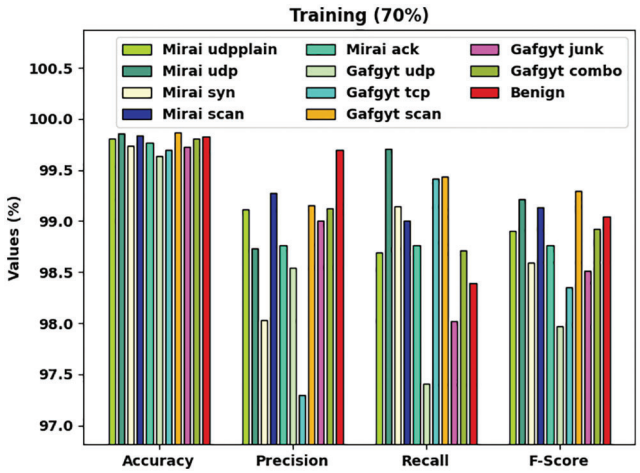

Figure 6: Classification outcomes of MFO-RELM model on 70% of training dataset

Fig. 7 illustrates the confusion matrix offered by the MFO-RELM model on 30% of testing dataset. The figure reported that the MFO-RELM model has identified 306 samples under Mirai udpplain, 296 samples under Mirai udp, 293 samples under Mirai syn, 298 samples under Mirai scan, 270 samples under Mirai ack, 302 samples under Gafgyt udp, 312 samples under Gafgyt tcp, 284 samples under Gafgyt scan, 291 samples under Gafgyt junk, 299 samples under Gafgyt combo, and 314 samples under Benign class.

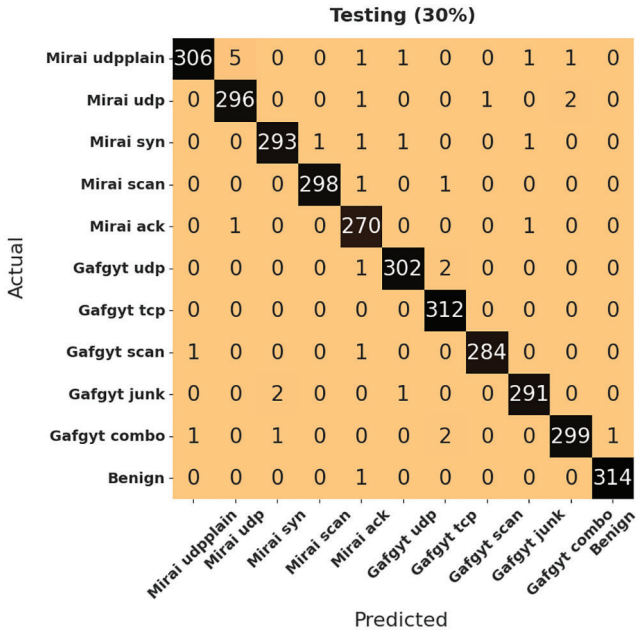

Figure 7: Confusion matrix of MFO-RELM model on 30% of testing dataset

Tab. 3 and Fig. 8 demonstrate extensive classification outcomes of the MFO-RELM model on 30% of training dataset. The results exhibited that the MFO-RELM model has gained maximum results in each class. For instance, in Mirai udpplain class, the MFO-RELM model has provided $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.67%, 99.35%, 97.14%, and 98.23% respectively. Simultaneously, in Mirai udp class, the MFO-RELM model has offered $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.70%, 98.01%, 98.67%, and 98.34% respectively. In line with, on Mirai syn class, the MFO-RELM model has reached $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.79%, 98.99%, 98.65%, and 98.82% respectively. Finally, in Mirai udpplain class, the MFO-RELM model has attained $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.91%, 99.67%, 99.33%, and 99.50% respectively.

Table 3: Classification results of MFO-RELM model on 30% of testing set

Testing set (30%)				
Class labels	Accuracy	Precision	Recall	F-score
Mirai udpplain	99.67	99.35	97.14	98.23
Mirai udp	99.70	98.01	98.67	98.34
Mirai syn	99.79	98.99	98.65	98.82
Mirai scan	99.91	99.67	99.33	99.50
Mirai ack	99.73	97.47	99.26	98.36
Gafgyt udp	99.82	99.02	99.02	99.02
Gafgyt tcp	99.85	98.42	100.00	99.21
Gafgyt scan	99.91	99.65	99.30	99.47
Gafgyt junk	99.82	98.98	98.98	98.98
Gafgyt combo	99.76	99.01	98.36	98.68
Benign	99.94	99.68	99.68	99.68
Average	99.81	98.93	98.95	98.94

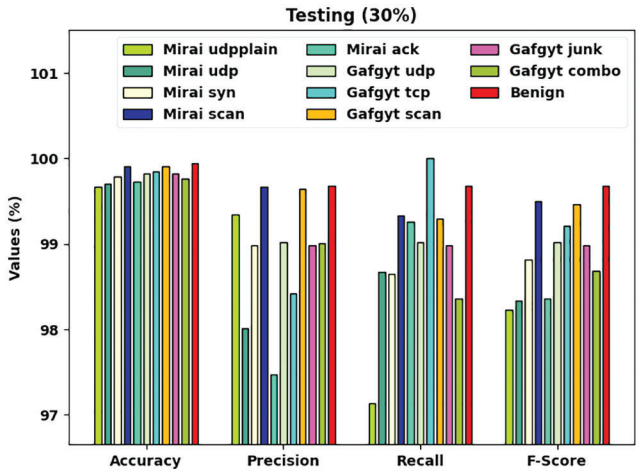

Figure 8: Classification outcomes of MFO-RELM model on 30% of testing dataset

Fig. 9 validates the training/validation accuracies gained by the MFO-RELM model on the test dataset. The results understood that the MFO-RELM model has resulted in maximum training/validation accuracies on the test data with a rise in epoch count.

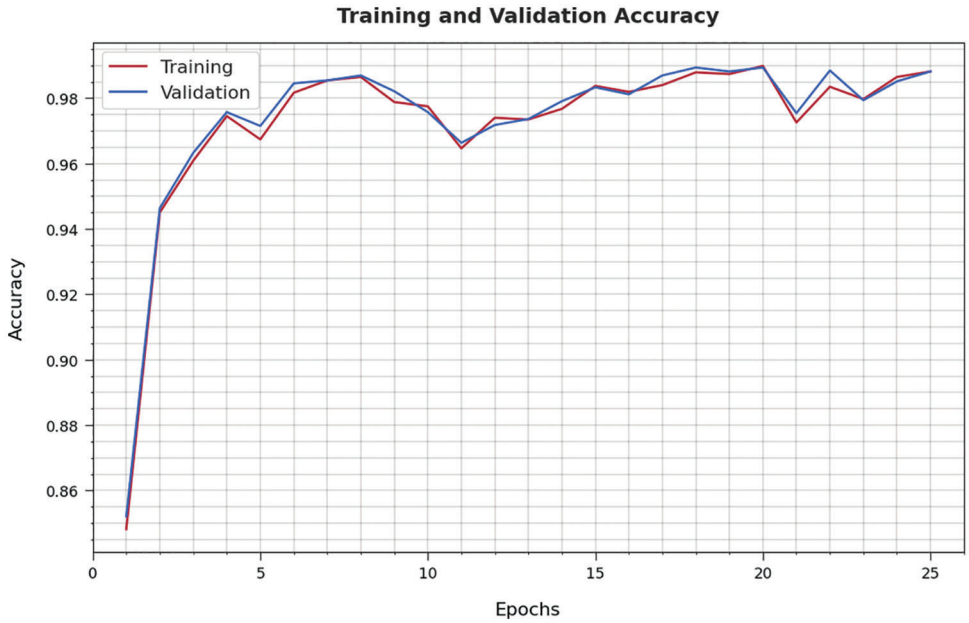

Figure 9: Training/validation accuracies of MFO-RELM model

Fig. 10 reports the training/validation losses conveyed by the MFO-RELM model on the test dataset. The results concluded that the MFO-RELM model has reached least training/validation losses on the test data with a rise in epoch count.

Figure 10: Training/validation losses of MFO-RELM model

A detailed comparative study of the MFO-RELM with other models [20] is made in Tab. 4 and Fig. 11. The experimental results implied that the MFO-RELM model has outperformed the other methods. For $prec_n$, the MFO-RELM model has offered increased $prec_n$ of 98.93% whereas the BA-NN, PSO-NN, and LGBA-NN models have resulted in reduced $prec_n$ values of 84.53%, 83.94%, and 87.34% respectively. Besides, for $reca_l$, the MFO-RELM model has presented enhanced $reca_l$ of 88.02% whereas the BA-NN, PSO-NN, and LGBA-NN models have resulted in reduced $prec_n$ values of 88.02%, 88.81%, and 92.42% respectively. Similarly, for $prec_n$, the MFO-RELM model has gained maximum F_{score} of 98.94% whereas the BA-NN, PSO-NN, and LGBA-NN models have reached minimal F_{score} values of 84.68%, 84.31%, and 89.18% respectively.

Table 4: Comparative study of MFO-RELM model

Methods	Precision	Recall	F-score
BA-NN model	84.53	88.02	84.68
PSO-NN model	83.94	88.81	84.31
LGBA-NN model	87.34	92.42	89.18
MFO-RELM	98.93	98.95	98.94

From the detailed results and discussion, it is ensured that the MFO-RELM model has accomplished maximum performance over the other recent methods.

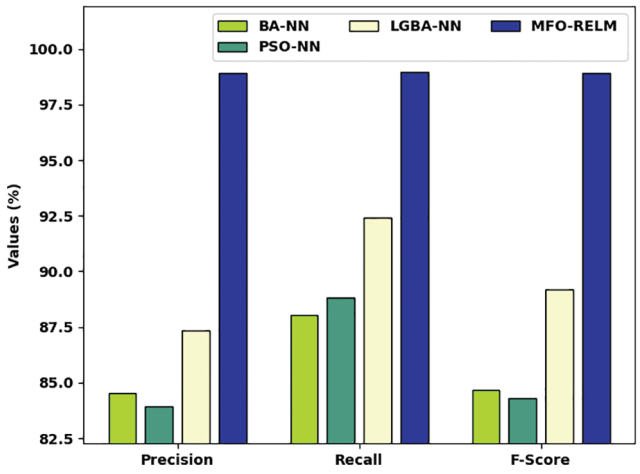

Figure 11: Comparative Classification outcomes of MFO-RELM model

4 Conclusion

In this study, a new MFO-RELM model is introduced for Cybersecurity Threat Detection and classification in IoT environment. The presented MFO-RELM technique accomplishes the effectual identification of cybersecurity threats that exist in the IoT environment. At the initial stage, the MFO-RELM model pre-processes the actual IoT data into a meaningful format. In addition, the RELM model receives the pre-processed data and carries out the classification process. To boost the performance of the RELM model, the MFO algorithm has been employed to it. The performance validation of the MFO-RELM model is tested using standard datasets and the results highlighted the better outcomes of the MFO-RELM model under distinct aspects. Thus, the MFO-RELM model is found to be effective in the recognition of cybersecurity threats in the IoT environment. In the future, novel clustering and outlier removal processes can be included to raise the efficiency of the MFO-RELM model.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under grant number (RGP 2/142/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R161), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4210118DSR06).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Salih, S. T. Zeebaree, S. Ameen, A. Alkhyyat and H. M. Shukur, "A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection," in *2021 7th Int. Engineering Conf. on Research & Innovation amid Global Pandemic (IEC)*, Erbil, Iraq, pp. 61–66, 2021.

[2] Z. Z. Xian and F. Zhang, "Image real-time detection using lse-yolo neural network in artificial intelligence-based internet of things for smart cities and smart homes," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–8, 2022.

[3] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke and L. Shu, "Federated deep learning for cyber security in the internet of things: Concepts, applications and experimental analysis," *IEEE Access*, vol. 9, pp. 138509–138542, 2021.

[4] Y. Li, Y. Zuo, H. Song and Z. Lv, "Deep learning in security of internet of things," *IEEE Internet of Things Journal*, pp. 1, 2021.

[5] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," *Computer Science Review*, vol. 39, no. 4, pp. 100317, 2021.

[6] D. Chen, P. Wawrzynski and Z. Lv, "Cyber security in smart cities: A review of deep learning-based applications and case studies," *Sustainable Cities and Society*, vol. 66, pp. 102655, 2021.

[7] R. Ahmad and I. Alsmadi, "Machine learning approaches to IoT security: A systematic literature review," *Internet of Things*, vol. 14, no. 3, pp. 100365, 2021.

[8] A. D. Raju, I. Y. Abualhaol, R. S. Giagone, Y. Zhou and S. Huang, "A survey on cross-architectural IoT malware threat hunting," *IEEE Access*, vol. 9, pp. 91686–91709, 2021.

[9] B. Jothi and M. Pushpalatha, "WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks," *Personal and Ubiquitous Computing*, vol. 82, no. 4, pp. 761, 2021.

[10] E. Bout, V. Loscri and A. Gallais, "How machine learning changes the nature of cyberattacks on IoT networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 1, pp. 248–279, 2022.

[11] X. L. Novo, V. A. Villagrá, M. V. Barbas, D. Rivera and M. S. Rodrigo, "An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets," *Sensors*, vol. 21, no. 2, pp. 656, 2021.

[12] N. Yadav, S. Pande, A. Khamparia and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1–3, pp. 1–13, 2022.

[13] S. M. Ali, A. S. Elameer and M. M. Jaber, "IoT network security using autoencoder deep neural network and channel access algorithm," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 95–103, 2021.

[14] M. Y. Alzahrani and A. M. Bamhdi, "Hybrid deep-learning model to detect botnet attacks over internet of things environments," *Soft Computing*, vol. 6, no. 164–175, pp. 2432, 2022.

[15] W. Xie, J. Wang, C. Xing, S. S. Guo, M. Guo *et al.,* "Adaptive hybrid soft-sensor model of grinding process based on regularized extreme learning machine and least squares support vector machine optimized by golden sine harris hawk optimization algorithm," *Complexity*, vol. 2020, no. 344, pp. 1–26, 2020.

[16] F. K. Inaba, E. O. T. Salles, S. Perron and G. Caporossi, "DGR-ELM-distributed generalized regularized ELM for classification," *Neurocomputing*, vol. 275, no. 2, pp. 1522–1530, 2018.

[17] G. Kannan, P. S. Kumar and V. P. Vinay, "Comments on 'nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm' by jung-bok jo, yinzhen li, mitsuo gen, computers & industrial engineering (2007)," *Computers & Industrial Engineering*, vol. 55, no. 2, pp. 533–534, 2008.

[18] X. Guo, X. Yan and Jermsittiparsert, "Using the modified mayfly algorithm for optimizing the component size and operation strategy of a high temperature PEMFC-powered CCHP," *Energy Reports*, vol. 7, no. 11, pp. 1234–1245, 2021.

[19] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai *et al.,* "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[20] A. Alharbi, W. Alosaimi, H. Alyami, H. T. Rauf and R. Damaševičius, "Botnet attack detection using local global best bat algorithm for industrial internet of things," *Electronics*, vol. 10, no. 11, pp. 1341, 2021.