IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 March 2024, accepted 7 April 2024, date of publication 17 April 2024, date of current version 10 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3390093

RESEARCH ARTICLE

Enhancing Cybersecurity in the Internet of Things Environment Using Artificial Orca Algorithm and Ensemble Learning Model

RANDA ALLAFI[1] AND IBRAHIM R. ALZAHRANI[2]

[1]Department of Computers and Information Technology, College of Sciences and Arts, Northern Border University, Arar 91431, Saudi Arabia
[2]Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, Al Jamiah, Hafar Al Batin 39524, Saudi Arabia

Corresponding author: Ibrahim R. Alzahrani (ialzahrani@uhb.edu.sa)

This work was supported by the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, through the research work under Project NBU-FFR-2024-170-04.

ABSTRACT Cybersecurity in the Internet of Things (IoT) is the practice of implementing measures to secure networks and connected devices from data breaches, cyber threats, and unauthorized access. It is essential owing to the increasing interconnectivity of devices, ranging from smart home appliances to industrial sensors. The potential attack surface expands, necessitating strong cybersecurity measures to protect sensitive data, ensure privacy, and prevent disruptions to critical services with these increasing number of IoT devices. Artificial intelligence (AI) technologies, particularly deep learning (DL) and machine learning (ML) approaches, hold the potential to mitigate and identify cyberattacks on IoT networks. DL demonstrates promise for effectively preventing and detecting security threats within IoT devices. Despite the importance of Intrusion Detection Systems (IDS) in maintaining confidentiality by detecting suspicious activities, classical IDS solutions might face difficulties in the IoT platform. Therefore, this study presents an Artificial Orca Algorithm with Ensemble Learning cyberattack detection and classification (AOAEL-CDC) methodology in an environment of IoT. The presented AOAEL-CDC technique exploited the feature selection (FS) approach with an ensemble learning approach for cyberattack recognition and identification in the IoT atmosphere. In the developed AOAEL-CDC model, the feature selection takes place using the AOA technique. For the cyberattack detection process, the ensemble learning process is carried out by the use of three models such as bidirectional long short-term memory (BiLSTM), gated recurrent unit (GRU), and extreme learning machine (ELM). Finally, the hyperparameter range of the DL techniques takes place using the marine predator's algorithm (MPA). To examine the performance analysis of the AOAEL-CDC methodology, a series of simulations take place using a benchmark dataset. An extensive comparative study reported that the BCODL-SDSC technique reaches an effective performance with other models with a maximum accuracy of 99.31%.

INDEX TERMS Cyberattack, Artificial Orca Algorithm, ensemble learning, hyperparameter tuning, deep learning.

I. INTRODUCTION

IoT is an effective system for interconnected devices like actuators and sensors which able to gather data at varying speeds. With uncertain sensors and consistent systems, data collection in reality state is proficient at the highest level [1]. Every day, IoT is growing rapidly, and assessed that IoT-based users will be nearly 20.4 billion. Owing to its outstanding features like automation, trustworthiness, scalability, and sturdiness, IoT has gained huge popularity [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

These features can convert the upcoming IoT uses and improve the quality of service (QoS) that is provided by IoT uses namely healthcare, smart cities, manufacturing automation, and smart transports [3]. However, the combination of IoT through numerous gadgets advances the safety anxieties in IoT uses. IoT connects additional devices via a central server which grows the confidentiality and safety anxieties [4]. The assorted and vibrant nature of IoT gadgets makes them inclined to dissimilar kinds of threats and safety assaults.

Conventional security models like access control, device confirmation, malware recognition, and cryptography-based approaches have been developed beforehand to enlarge the security of IoT [5]. However, the classification of dissimilar kinds of cyber threats and safety threats employing these kinds of models is a challenging task. Besides, numerous uncertain and dangerous operational issues enhance the security threats that weaken the reliability of the IoT model [6]. Present safety methods must be converted to identify new cyber threats. It needs a smart and intelligent technique to detect various kinds of threats in IoT namely distributed DoS (DDoS), jamming, denial of service (DoS), flooding, and botnet. Artificial intelligence (AI) based models have been utilized in the design and improvement of a clever attack recognition technique for safeguarding IoT devices [7]. The most suitable use of AI model, particularly machine learning (ML) will aid the investigators in identifying anomalies or unnecessary malicious actions in the IoT. As an outcome, it provides a dynamic safety solution that will regularly improve. Rule-based IDS models have been traditionally utilized but the difficulty and variety of IoT systems have reduced their effectiveness [8]. Deep learning (DL) is one of the techniques that could be utilized to increase the effectiveness and accuracy of IDS for IoT. To increase the usage of IDSs in IoT devices using DL and further detect the faults and powers of these systems, it is highly vital to assess the obtainable literature and magazines widely [9]. Particularly, ML or DL techniques include a set of regulations, approaches, or difficult relocation functions that remove beneficial insights or appealing data designs from the safety facts. Therefore, it is highly probable to employ the subsequent safety methods for training machines to forecast dangers or threats at a primary phase [10].

In the context of the Internet of Things (IoT), existing cybersecurity approaches face an abundance of difficulties that degrade the efficiency of protecting IoT environments. One remarkable complexity lies in the sheer heterogeneity of IoT devices and transmission protocols, resulting in different data structures and formats. Current cybersecurity techniques often encounter challenges in adapting to this heterogeneity, limiting their capability to mitigate and detect threats across a considerable degree of devices. Furthermore, the resource-constraint nature of IoT devices makes it challenging to implement strong security measures, as classical security protocols may be too computationally intensive for these devices. Furthermore, the real-time and dynamic nature of IoT environments demands models that can accurately and quickly respond to emerging threats, yet several existing techniques lack the agility required for quick adaptation. Moreover, the scalability problem arises as IoT environments continue to extend, putting pressure on cybersecurity approaches to powerfully handle the complexity and increasing volume of data without adversely affecting performance. Resolving these problems is crucial for advancing cybersecurity in IoT, requiring new techniques that account for the diverse and dynamic nature of IoT deployments.

To overcome these challenges, hyperparameter tuning and ensemble learning can be combined with Cybersecurity in the IoT is grounded in the difficult impediments posed by the heterogeneous and dynamic nature of IoT ecosystems. Ensemble learning, with its capability to combine the strength of multiple models, overcomes the limitation of individual ML techniques in evolving patterns and capturing the diverse cyberattacks within IoT environments. The fundamental conflicts of IoT, marked by different data sources, device types, and transmission protocols, demand a more adaptable and robust defense mechanism. Ensemble learning provides a potential avenue to enhance the overall accuracy, resilience, and generalization of cybersecurity algorithms in IoT by leveraging various perspectives and improving the collective intelligence of the system. At the same time, hyperparameter tuning is motivated by the need for fine-tuning ML techniques to align with the intricacies of cybersecurity and IoT data requirements. The massive variability in IoT deployments demands a nuanced adjustment of model configurations to improve performance. Hyperparameter tuning ensures that the models are finely calibrated to certain features of IoT data, considering factors concluding data diversity, volume, and the real-time nature of attacks.

This research develops an Artificial Orca Algorithm with Ensemble Learning-based cyber threat detection and classification (AOAEL-CDC) methodology in the IoT environment. The presented AOAEL-CDC technique exploited the feature selection (FS) approach with an ensemble learning approach for cyberattack detection and classification in the IoT atmosphere. In the developed AOAEL-CDC technique, the feature selection takes place employing the AOA technique. For cyberattack detection, the ensemble learning method is carried out by the use of three techniques namely bidirectional long short-term memory (BiLSTM), gated recurrent unit (GRU), and extreme learning machine (ELM). Finally, the hyperparameter selection of the DL techniques takes place by employing the marine predator's algorithm (MPA). To examine the performance study of the AOAEL-CDC model, a series of simulations take place employing a benchmark dataset.

II. RELATED WORKS

In [11], a firstly influences ML and DL techniques for the exact removal of vital features from a real-network traffic dataset of BoT-IoT. Then, the technique measures the efficiency of 10 different ML techniques in identifying malware.

This study contains 2 single classifiers (KNN and SVM), 8 ensemble classifiers namely Extra Trees, AdaBoost, and LGBM, as well as 4 DL architectures such as LSTM, GRU, and RNN. In [12], a method has been proposed by executing an innovative DL model to identify cyber-attacks besides IoT methods. Specifically, the developed model combines LSTM units into a joint of detectors. Then, these units are fused by utilizing a decision tree (DT) to reach a combined output at the concluding phase.

Alattas and Mardani [13] project a novel structure design that conveys a stochastic dimension of defining limits depending on a new adaptive deep learning (ADL) technique for trade manufacturers like transportation. Then, a new structure will be intended to include a component of arbitrary conflict to define outlines. The executed technique measured the system of forensic and intrusion detection systems (IDs) and is dependent on five selected protection phases. Maghrabi et al. [14] focus on the strategy of the Golden Jackal Optimizer with DL-based Cyber Threat Detection and Classification (GJODL-CADC) methodology in the network of IIoT. The technique uses a GJO-based feature selection method for classification. Then, the GJODL-CADC methodology utilizes a hybrid auto-encoder-based deep belief network (AE-DBN) model. The efficacy of the technique can be enhanced via a pelican optimizer algorithm (POA).

In [15], the attack recognition method has been developed for IoT employing Software-defined networks (SDNs). The SDNs can able to analyze the traffic movement, identify the anomaly, and block external traffic and source nodes. A Fuzzy neural network (FNN) based threat recognition method has been considered which is capable of discovering attacks like middleman, DDoS, side-network, and mischievous code. Bhattacharjee et al. [16] aim is to progress a Convolutional Neural Networks (CNNs)-based IDS to enhance the security of the internet. The suggested IDS design categorizes all system packet traffic into forms that are kind or mischievous to classify network intrusions. For the recommended method, CNNs, DNN, Logistic Regression, Adaboost, and RF four significant experimental DL methods have been taken into attention.

In [17], an integrated DL methodology is proposed utilizing Hybrid Dual-Channel CNN (DCCNN) with Spider Monkey Optimizer (SMO) model namely DCCNN-SMO. Tawfik et al. [18] developed a method by signifying middleware. ML technique has comprised in the middle-ware to deliver automatic defense besides cyber-attacks on IoT systems. A promising technique to protect actual, extremely exact assaults on SDN-managed IoT systems was developed. In [19], an innovative model implemented a hybrid deep learning model, combining Graph Convolutional Long Short-Term Memory (GC-LSTM) and a deep convolutional network is presented. Nanjappan et al. [20] propose DeepLG SecNet, an innovative strategy that utilizes a blend of deep learning methodologies, such as Long Short-Term Memory (LSTM) and gated Secure Network (SecNet), along with Crossover Chaos Game Optimization (CCGO) techniques.

There exists a conspicuous research gap about the effective integration of ensemble learning approaches. While individual ML techniques were utilized for threat mitigation and anomaly detection in IoT ecosystems, the dynamic and complex nature of IoT information often results in varied patterns that may go unnoticed by the single model. Ensemble learning, which integrates the predictions of multiple models, has illustrated its efficiency in optimizing overall detection robustness and accuracy in different domains. But the limited attention has been paid to exploring the ensemble techniques for improving cybersecurity measures in IoT environments. Addressing this gap includes developing new ensemble learning approaches tailored to the unique challenges confronted by IoT security, involving the heterogeneity of IoT devices, the variability in data sources, and the real-time nature of threats. Moreover, another crucial research gap in the field of IoT cybersecurity lies in the insufficiency of hyperparameter tuning methods. Hyperparameter tuning is essential to optimize the performance of ML approaches, yet existing study often overlooks the nuanced requirement of IoT ecosystems. The wide variety in data characteristics, IoT deployments, and transmission protocols requires a tailored technique to hyperparameter tuning that considers the specificities of IoT cybersecurity issues.

III. THE PROPOSED MODEL

In this study, we have presented an AOAEL-CDC technique on the IoT environment. The presented AOAEL-CDC technique exploited the feature selection (FS) approach with an ensemble learning approach for cyberattack recognition and classification in the IoT environment. Fig. 1 demonstrates the entire procedure of the AOAEL-CDC technique.

A. AOA FEATURE SELECTION

In the presented AOAEL-CDC technique, the feature selection takes place by employing the AOA technique. The Swarm Intelligence (SI) Algorithm has gained more attention in recent years [21]. The authors developed these procedures imitating orcas in their existing atmosphere. Presently, to include the cultural measurement of orcas, AOA has been crossed by the cultural algorithm (CA) to progress a system termed OCA. The orca's social organization contains numerous clans comprising in their turn pods. All these occurrences are performed in an algorithm termed the Artificial Orcas algorithm (AOA). This algorithm is monitored by a ruler which is measured as the best individual in the pod. Additionally, every hierarchical structure stage is prominent by a grade of nearness. The pods are nearer to themselves when compared to the clans. AOA is presented by creating an outstanding stability among dual significant stages of evolutionary procedures namely search diversification and search intensification.

$$f_{group} = f_{\min} + (f_{\max} - f_{\min}) \tag{1}$$

$$v_{p_i}^{t} = v_{p_i}^{t-1} + f_i \times D_p + f_c \times D_c + f_{pop} \times D_{pop} \tag{2}$$

$$x_{p_i}^{t} = x_{p_i}^{t-1} + v_{p_i}^{t} \tag{3}$$

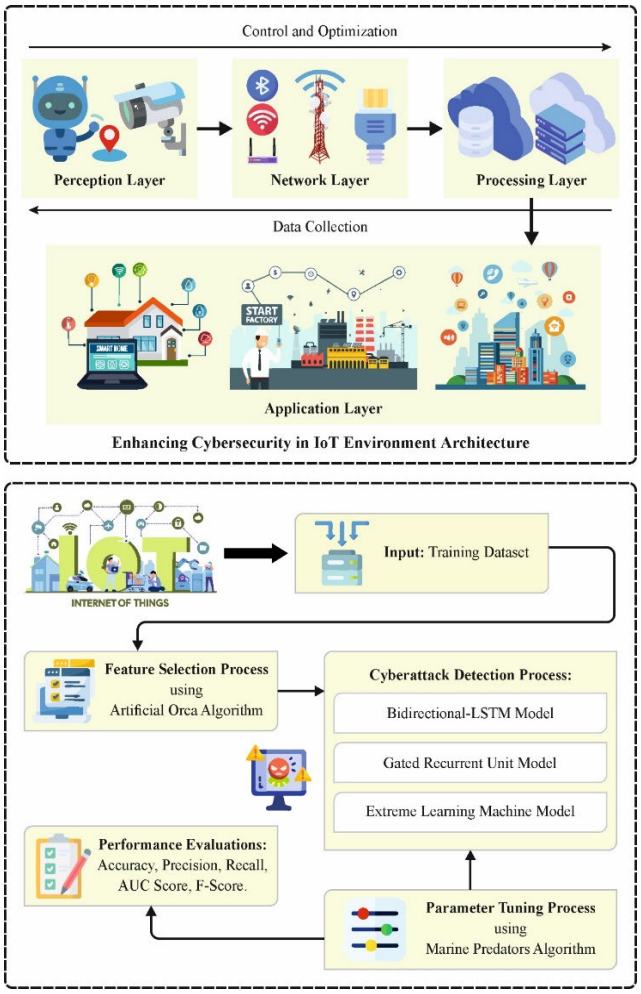

FIGURE 1. The overall process of the AOAEL-CDC technique.

$$x^{t}_{temp,p_i} = A \times \sin\left(\frac{2 \times \Pi}{L} \times x^{t-1}_{p_i}\right) \times \cos\left(\frac{2 \times \Pi}{T} \times t_x\right) \tag{4}$$

$$x^{t}_{m,p} = \frac{\sum_{j=1}^{j=n} x_{temp,p_j}}{n} \tag{5}$$

$$x^{t}_{p_i} = x^{*}_{p} - \beta \times x^{t}_{m,p} \tag{6}$$

$$x^{t}_{new,p_i} = \frac{\gamma \times x_{pop_{rand1}} + \omega \times x^{t}_{c_{rand2}}}{2} \tag{7}$$

From the above-mentioned equation, t denotes the existing iteration. $x^{t}_{p_i}$ represents the individual at position i in the pod $p.v^{t}_{p_i}$ signifies the speed of the individual at position $i.f_{\min}$ and $f_{\max}$ denotes the minimal and maximal frequencies, respectively, and are employed to make a random frequency f_{group}. It is noticeable that the cluster relates either to p, c, or pop to define the frequency. α, γ, and ω are said to be random numbers in interval [0,1]. The populace levels are correspondingly detached by distances of D_p, D_c, and D_{pop} that are definite as below.

$-D_p = \left|x^{t-1}_{p_i} - x^{*}_{p}\right|$ whereas x^{*}_{p} is the matriarch of the pod to which $x^{t-1}_{p_i}$ belongs.

$-D_c = \left|x^{t-1}_{p_i} - x^{*}_{c}\right|$ where x^{*}_{c} is the matriarch of the clan to that $x^{t-1}_{p_i}$ belongs.

$-D_{pop} = \left|x^{t-1}_{p_i} - x^{*}_{pop}\right|$ where x^{*}_{pop} is the matriarch of the population.

For the strategy of hunting, A is a parameter based on the problem exhibiting, L is a parameter demonstrating the wave length and T is an experimental parameter that signifies the wave period at the time of chasing action. x^{t}_{new,p_i} denotes the novel solution of the individual i in the pod $p.x^{t}_{pop,rand1}$ signifies the arbitrary individual, $rand1$ and $x^{t}_{c,rand2}$ is an arbitrary individual in the clan c of the present pod at location $rand2$.

Analysis of AOA Complexity: TC_{AOA} denotes the operations count offered in Eq. (8) based on the maximal iterations count Max$Iter$, n is the population size and #pods is a pods count. It is noticeable that the sorting technique utilized in the system is heapsort and its computation complexity is $O(nlogn)$.

$$TC_{AOA} = \sum_{i=1}^{\text{Max}Iter} (ntogn + n + \#pods + n) \tag{8}$$

$nlogn$ operation is required for Instruction 3, n operation for Instructions 4 and 6, and #pods operation for Instruction 5. As the dimension of the population $n = \#clans \times \#pods \times \#orcas$, $n > \#pods$, we determine that:

$TCAOA < \sum_{i=1}^{MaxIter}$ (nlogn+3n) and so the AOA computation complexity is $O(\text{Max}Iter \times nlogn)$

In the AOA methodology, the goals are combined into a single main formulation such that a present weight classifies every objective position [22]. In this manuscript, we assume a fitness function (FF) that integrates both goals of FS as displayed in Eq. (9).

$$Fitness\,(X) = \alpha \cdot E\,(X) + \beta * \left(1 - \frac{|R|}{|N|}\right) \tag{9}$$

where Fitness(X) indicates the fitness value of a subset X, $E(X)$ denotes the error rate of classification by using the nominated features, $|R|$ and $|N|$ are the selected count features and original features count respectively, α and β represents the measures of the classification error and decline ratio, correspondingly, where $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$.

B. ENSEMBLE LEARNING PROCESS

For the cyberattack detection process, the ensemble-learning process is carried out by the use of three models namely BiLSTM, GRU, and ELM.

1) BiLSTM MODEL

LSTM neural networks are a simulated DL model that depends upon recurrent neural networks (RNNs) [23]. This method disables the vanishing gradient issue of RNNs. This

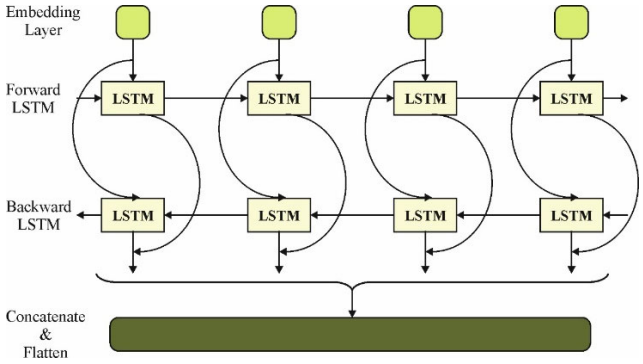

FIGURE 2. Architecture of BiLSTM.

method has been commonly employed for time sequence data forecasts and attained outstanding outcomes.

It contains memory cells, memory blocks, and gate units. A cell state keeps data. The technique utilizes these kinds of gates to keep and procedure the related data. These gates will learn what data must be maintained and then ignored. An input gate defines which data must be inserted into a cell state. The output gate offers outputs. A forget gate defines which data should be maintained from a preceding layer. Fig. 2 depicts the infrastructure of BiLSTM.

During this study, a BiLSTM method based upon conventional LSTM neural networks has been projected to forecast hydroelectric power depending on multi-variable inputs. The BI-LSTM reflects past and upcoming layers to enhance forecast accuracy. Whereas normal LSTMs reflect only past states, BiLSTM reflects both future and past states. Converse LSTMs utilize upcoming data and forward LSTMs utilize past data. The BiLSTM attains superior accurateness when compared to LSTM because it uses past as well as future data. Eqs. (10) to (15) arithmetically define the connection among weighted outputs and inputs:

$$O_t = \sigma\left(\varphi_o z_{t-1} + V_o x_t + \kappa_o\right) \tag{10}$$
$$I_t = \sigma\left(\varphi_i z_{t-1} + V_i x_t + \kappa_b\right) \tag{11}$$
$$F_t = \sigma\left(\varphi_f z_{t-1} + V_f x_t + \kappa_f\right) \tag{12}$$
$$\hat{S}_t = tanh\left(\varphi z_{t-1} + V x_t + b\right) \tag{13}$$
$$S_t = F_t \odot S_{t-1} + i_t \odot \hat{S}_t \tag{14}$$
$$z_t = O_t \odot tanh\left(S_t\right) \tag{15}$$

where F_t, O_t, *and* I_t denote the forget output and input gates, x_t and z_t signify an input and output state at time t, S_t represents the memory cell, $\hat{S}_t$ refers to the novel value of memory cells, φ_i, φ_o, and φ_f denotes the weight matrixes of the hidden layer (HL), and V_o, V_i, V_f represents the weights consistent with input data, and $f, I, 0$: corresponds to the forget, input, and output gate.

A BiLSTM system contains both forward and backward LSTMs which can route the data in both ways. In forward, computations are executed from time 1 to t. While in backward, calculation executes from time t to 1.

2) GRU MODEL

GRU is a novel structure prepared to find out the problem of vanishing or exploding gradient [24]. The updated structure of LSTM is termed GRU. To control the data movement, GRUs too have a gate structure like LSTM. But, GRU wants an output gate, permitting the content to be completely open. The GRU has only dual gates such as reset and update gates. The forget and input gates of the LSTM structure are unified into the 2nd gate. When equated with LSTM, GRUs have a simple framework and few limits, which will enhance the solution.

The GRU formula is expressed below:

$$r_t = sigm\left(W_{xr} x_t + W_{hr} h_{t-1} + b_r\right) \tag{16}$$
$$z_t = sigm\left(W_{xz} x_t + W_{hz} h_{t-1} + b_z\right) \tag{17}$$
$$\tilde{h}_t = tanh\left(W_{xh} x_t + W_{hh}\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{18}$$
$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot \tilde{h}_t\right) \tag{19}$$

where x denotes the input vector and h_t is the output vector, r_t signifies the reset gate and z_t refers to the update gate. Parallel to LSTM, 'b' refers to biases, and 'W' for weight. While *sigm* denotes the activation sigmoid function and *tanh* refers to the tangent function. Both structures of LSTM and GRU can able to handle the long dependency. But, there are few variants in terms of performance. In this research, we used both structures to estimate how well they categorized network traffic.

3) ELM MODEL

The fundamental components of ELM comprise HL, input, and output layers. During the training period, the HL neuron is not to be optimized [25]. This neuron can never refurbished and arbitrarily distributed. Input is connected to HL with random weight (w_i) and the bias (b_j) whereby computational and time complexities are undermined compared to BPNN and ANN. When compared to BPNN, ELM is faster. In comparison with gradient-based approaches, Non-gradient-based ELM has the best generalization performance and prevents overfitting, local minima, and invalid learning rates. The output functions of ELM with hidden nodes (L) for the training set $R = \{(X_i, t_i)\}$, $i = 1, 2, \ldots, n$ are given below:

$$f\left(R\right) = \sum_{j=1}^{L} \beta_j H\left(X\right) = t_j \tag{20}$$

In Eq. (20), the weight matrices amongst HLs and output layers are represented as $\beta = \beta_1, \beta_2, \ldots, \beta_L$, and the target matrix of the training dataset is denoted by $r = r_1, r_2, \ldots, r_L$. The output of HL is determined by using Eq. (21):

$$H = \begin{bmatrix} G(w_1, b_1, X_1) & \cdots & G(w_L, b_L, X_1) \\ G(w_1, b_1, X_2) & \cdots & G(w_L, b_L, X_2) \\ \vdots & \vdots & \vdots \\ G(w_1, b_1, X_n) & \cdots & G(w_L, b_L, X_n) \end{bmatrix}$$
$$\beta = (H^T H)^{-1} H^T \tag{21}$$

The activation functions concerning weight, bias, and input are represented as 'G'. The activation functions adopted are Gaussian, Sigmoidal, Fourier series, and Hard limit functions. Here, a sigmoidal function is assumed as an activation function.

$$G(w_i, b_i, X_i) = \frac{1}{1 + e^{-(wx+b)}} \tag{22}$$

The desired output of ELM can be defined as.

$$T_{test} = H\beta \tag{23}$$

C. MPA-BASED HYPERPARAMETER TUNING

Finally, the hyperparameter selection of the DL models takes place utilizing MPA. MPA has been proposed meta-heuristic methodology that imitates the relationship between prey and predator [26]. MPA's foremost goal is to search for food, whereas a hunter closely hunts for food and prey. MPA algorithm was proposed by considering predator as well as prey as the best solution. This algorithm begins with an initialized stage and then passes by other 3 stages with esteem to the normal speed between predator and prey.

- Initialize stage: This stage offers an arbitrary set of solutions for prey as well as predator through the following formulation:

$$U = Lower + rand_1 \times (UpperLower) \tag{24}$$

where the Upper denotes the upper and Lower signifies the lower bound in the hunt space, $rand_1$ refers to a random vector $\in$ *the* interval of (0, 1). Affording to the above formulation, the early positions of predator and prey can be determined as follows:

$$Elite = \begin{bmatrix} U_{11}^1 & U_{12}^1 & \cdots & U_{1d}^1 \\ U_{21}^1 & U_{22}^1 & \cdots & U_{2d}^1 \\ \cdots & \cdots & \cdots & \cdots \\ U_{n1}^1 & U_{n2}^1 & \cdots & U_{nd}^1 \end{bmatrix},$$

$$U = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1d} \\ U_{21} & U_{22} & \cdots & U_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ U_{n1} & U_{n2} & \cdots & U_{nd} \end{bmatrix}, \tag{25}$$

whereas the Elite matrix denotes the best predator.

- **Stage 1**: The exploration stage has been executed to determine the search space after the beginning. So in MPA, for the 1st third of the complete iterations, (i.e., $\frac{1}{3}t_{\max}$). Therefore the prey location is updated which depends upon the following equations.

$$S_i = R_B \otimes (Elite_i - R_B \otimes U_i), i = 1, 2, \ldots, n \tag{26}$$

$$U_i = U_i + P.R \otimes S_i \tag{27}$$

whereas $R \in [0, 1]$ signifies the arbitrary vector drawn from an even distribution and $P = 0.5$ refers to a constant number. R_B denotes to motion of Brownian. $\otimes$ designates the procedure of unit-wise multiplication.

- **Stage 2**: The predator or prey starts using the finest place that identifies for their foods. Phase 2 is implemented in the 2nd third of the complete iterations count when $\frac{1}{3}t_{\max} < t < \frac{2}{3}t_{\max}$. It separated the agents for dual splits and expressed in Eqs. (28) and (29) to match the motion of the 1st half (prey) and the 2nd half (predator) is expressed in Eqs. (32) and (33) as below.

$$S_i = R_L \otimes (Elite_i - R_L \otimes U_i), i = 1, 2, \ldots, n/2 \tag{28}$$

$$U_i = U_i + P.R \otimes S_i \tag{29}$$

whereas R_L has arbitrary numbers that obey Lévy distribution. Eq. (28) and (29) are executed in the 1st half which signifies the exploitation. Whereas the 2nd half does the following expressions.

$$S_i = R_B \otimes (R_B \otimes Elite_i - U_i), i = 1, 2, \ldots, n/2 \tag{30}$$

$$U_j = Elite_i + P.CF \otimes S_j, CF = (1 - \frac{t}{t_{max}})^{\left(2\frac{t}{t_{max}}\right)} \tag{31}$$

where CF denotes the parameter that manages the step size of flow for the predator.

- **Stage 3**: This phase implemented on the last third of the iteration counts ($t > \frac{2}{3}t_{\max}$) depends upon the following formulation:

$$S_i = R_L \otimes (R_L \otimes Elite_i - U_i), i = 1, 2, \ldots, n \tag{32}$$

$$U_i = Elite_i + P.CF \otimes S_i, CF = (1 - \frac{t}{t_{max}})^{\left(2\frac{t}{t_{max}}\right)} \tag{33}$$

- Fish Aggregating Devices and EddyFormation Effect: Exterior effects from the atmosphere, like Fish Aggregating Devices (FADs) or eddy formation effects, are measured to evade the local optimal solution. It is executed as below:

$$U_i = \begin{cases} U_i + CF\,[U_{\min} + R \otimes (U_{\max} - U_{\min})] \otimes Wr_5 < FAD \\ U_j + [FAD(1-r) + r](U_{r1} - U_{r2})r_5 > FAD \end{cases} \tag{34}$$

From the above-mentioned expression, $FAD = 0.2$, and W is a dual solution (0 or 1) that parallels to random solution. If it is fewer than 0.2, then it is transformed to 0 whereas the arbitrary solution converts 1 when the solution is higher than 0.2. The $r \in [0, 1]$signifies the random number r_1, and r_2 denotes the random index.

- Marine memory: Marine hunters are the main feature and aid in holding an optimum solution quickly as well as preventing local solutions. It keeps preceding the best solution of a previous iteration, and evaluating by present ones; the solution is changed depending upon the finest one during the comparison step.

The fitness selection is the significant factor manipulating the solution in the MPA technique. The hyperparameter selection procedure includes the solution encode method to assess the efficiency of candidate solutions. During this study,

TABLE 1. Details on dataset.

Classes	No. of Instances
Normal	1000
Fuzzers	1000
DoS	1000
Analysis	1000
Exploits	1000
Generic	1000
Total Instances	6000

MPA methodology reflects accuracy as the foremost norm for designing the FF, which is expressed below.

$$Fitness = \max(P) \tag{35}$$

$$P = \frac{TP}{TP + FP} \tag{36}$$

From the above-mentioned formulation, FP and TP represent the false and true positive values.

IV. PERFORMANCE VALIDATION

The proposed model is simulated using the Python 3.8.5 tool. The proposed model is experimented on PC i5-8600k, GeForce 1050Ti 4GB, 16GB RAM, 250GB SSD, and 1TB HDD. The performance validation of the AOAEL-CDC methodology has been verified using the UNSW dataset [27], including 6000 instances and 6 classes as described in Table 1. Among the obtainable 49 features, the AOAEL-CDC model has been nominated for 28 features.

Fig. 3 displays the classifier outcomes of the AOAEL-CDC system below the test dataset. Figs. 3a-3b represents the confusion matrix acquired by the AOAEL-CDC methodology at 70:30 of TRPH/TSPH. This figure signified that the AOAEL-CDC method can be precisely recognized and categorized with 6 class labels. Next, Fig. 3c displays the PR study of the AOAEL-CDC algorithm. The figure described that the AOAEL-CDC technique attains excellent PR analysis in every class. Lastly, Fig. 3d authorizes the ROC study of the AOAEL-CDC methodology. The figure shows that the AOAEL-CDC technique offers effective results with improved ROC values below diverse classes.

Table 2, the overall cyberattack detection analysis of the AOAEL-CDC system with 70:30 of TRPH/TSPH. Fig. 4 illustrates the classifier analysis of the AOAEL-CDC technique on 70% of TRPH. The simulation values inferred that the AOAEL-CDC model has effective detection under six classes. With normal class, the AOAEL-CDC technique has obtained $anaccu_y$ of 99.52%, $prec_n$ of 99.14%, $reca_l$ of 98.01%, F_{score} of 98.57%, and AUC_{score} of 98.92%. In addition, in the DoS class, the AOAEL-CDC system provides $accu_y$ of 99%, $prec_n$ of 96.51%, $reca_l$ of 97.36%, F_{score} of 96.93%, and AUC_{score} of 98.34%. Moreover, based on generic class, the AOAEL-CDC method attains $anaccu_y$ of 99.10%, $prec_n$ of 97.22%, $reca_l$ of 97.22%, F_{score} of 97.22%, and AUC_{score} of 98.34%, correspondingly.

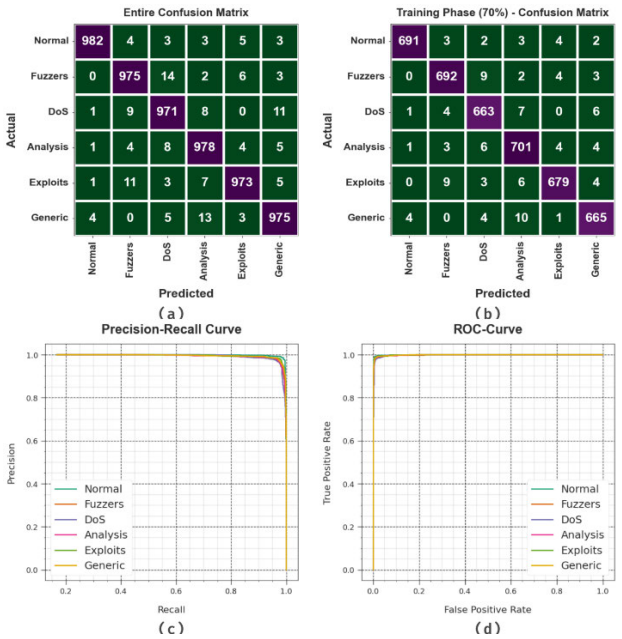

FIGURE 3. (a-b) Confusion matrices of 70:30 TRPH/TSPH (c) PR curve and (d) ROC curve.

TABLE 2. Cyberattack detection analysis of AOAEL-CDC model under 70:30 of TRPH/TSPH.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}	AUC_{score}
TRPH (70%)					
Normal	99.52	99.14	98.01	98.57	98.92
Fuzzers	99.12	97.33	97.46	97.40	98.46
DoS	99.00	96.51	97.36	96.93	98.34
Analysis	98.90	96.16	97.50	96.82	98.35
Exploits	99.17	98.12	96.86	97.49	98.25
Generic	99.10	97.22	97.22	97.22	98.34
Average	**99.13**	**97.41**	**97.40**	**97.41**	**98.44**
TSPH (30%)					
Normal	99.72	99.66	98.64	99.15	99.29
Fuzzers	99.11	96.92	97.59	97.25	98.50
DoS	98.89	97.16	96.55	96.86	97.97
Analysis	99.50	98.23	98.58	98.40	99.12
Exploits	99.44	98.33	98.33	98.33	99.00
Generic	99.22	97.48	98.10	97.79	98.78
Average	**99.31**	**97.96**	**97.96**	**97.96**	**98.78**

Fig. 5 shows the classifier analysis of the AOAEL-CDC method with 30% of TSPH. The experimental findings displayed that the AOAEL-CDC methodology can be an efficient detection of six classes. According to normal class, the AOAEL-CDC technique gets $anaccu_y$ of 99.72%, $prec_n$ of 99.66%, $reca_l$ of 98.64%, F_{score} of 99.15%, and AUC_{score} of 99.29%. Similarly, with the DoS class, the AOAEL-CDC system obtains $accu_y$ of 98.89%, $prec_n$ of 97.16%, $reca_l$ of 96.55%, F_{score} of 96.86%, and AUC_{score} of 97.97%. Besides,

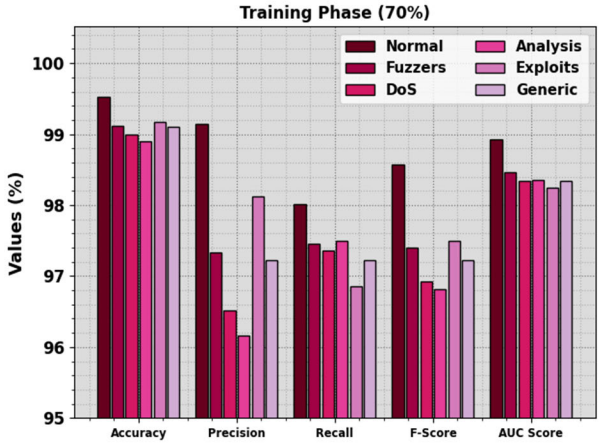

FIGURE 4. Cyberattack detection analysis of AOAEL-CDC system on 70% of TRPH.

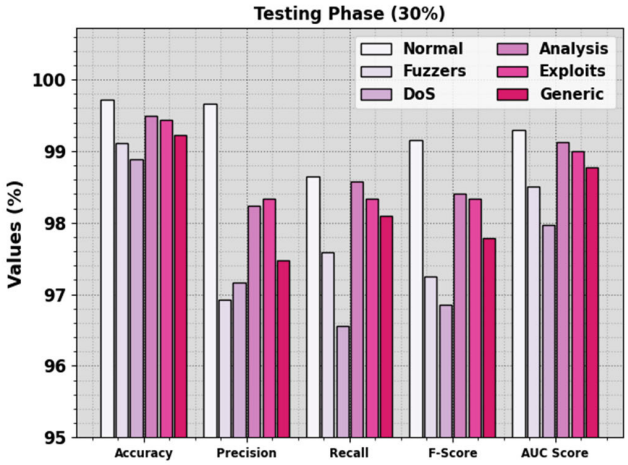

FIGURE 5. Cyberattack detection analysis of the AOAEL-CDC model under 30% of TSPH.

FIGURE 6. $Accu_y$ curve of the AOAEL-CDC algorithm.

on generic class, the AOAEL-CDC algorithm offers $accu_y$ of 99.22%, $prec_n$ of 97.48%, $reca_l$ of 98.10%, F_{score} of 97.79%, and AUC_{score} of 98.78%, respectively.

The $accu_y$ curves for training (TR) and validation (VL) displayed in Fig. 6 for the AOAEL-CDC algorithm provide valued insights into its effectiveness on diverse epochs.

FIGURE 7. Loss curve of the AOAEL-CDC model.

TABLE 3. Comparison analysis of the AOAEL-CDC system with other algorithms.

Technology	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
GA-LR	81.46	83.06	85.96	86.02
TS-RF	83.17	83.34	83.69	85.12
LSO-FNN	95.51	94.11	94.06	96.01
SCM3-RF	95.94	94.13	94.28	93.93
RHF-ANN	97.65	95.67	96.04	96.78
EAFS-RF	98.41	94.14	95.45	97.07
BHPO-MLPAD	99.14	97.44	97.40	97.42
AOAEL-CDC	99.31	97.96	97.96	97.96

Mainly, it can be a constant upgrading in both TR and TS $accu_y$ with increased epochs, representing the proficiency of the model to learn and recognize the patterns at both data of TR and TS. The improving trend in TS $accu_y$ underscores the adaptability of the model to the TR dataset and its ability to produce correct predictions on unnoticed data, emphasizing the capabilities of robust generalization.

Fig. 7 exhibits a wide-ranging overview of the TR and TS loss values for the AOAEL-CDC technique in different epochs. The TR loss reliably diminishes as the model refines its weights to reduce the classification error rate under both datasets. The loss curves demonstrate the model's alignment with the TR data, underscoring its ability for effectively capturing patterns. Significant is the incessant enhancement of parameters in the AOAEL-CDC system, targeted at minimalizing discrepancies among predictions and actual TR labels.

The comparison analysis of the AOAEL-CDC technique is demonstrated in Table 3 and Fig. 8 [28]. The acquired consequence implies the GA-LR and TS-RF methods demonstrate poorer performance. Simultaneously, the LSO-FNN and RHF-ANN algorithms illustrate relatively boosted results. Meanwhile, the RHF-ANN, EAFS-RF, SCM3-RF, and BHPO-MLPAD models depict reasonable results. However, the AOAEL-CDC technique exhibits better performance with a greater $accu_y$ of 99.31%, $prec_n$ of 97.96%, $reca_l$ of 97.96%, and F_{score} of 97.96%.

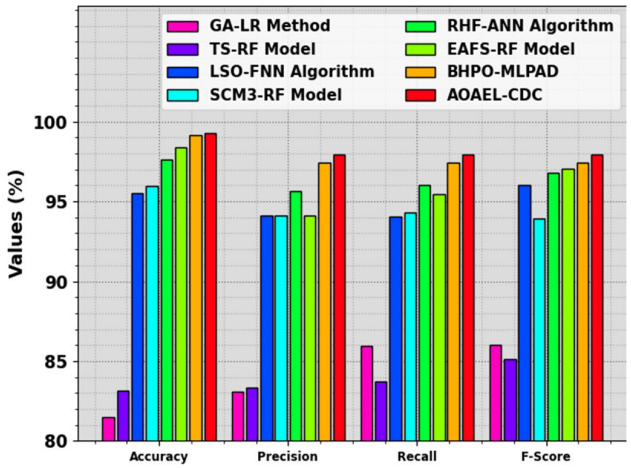

FIGURE 8. Comparison analysis of the AOAEL-CDC model with other techniques.

TABLE 4. CT analysis of the AOAEL-CDC model with other systems.

Technology	Computational Time (sec)
GA-LR Method	1.58
TS-RF Model	1.67
LSO-FNN Algorithm	1.63
SCM3-RF Model	1.81
RHF-ANN Algorithm	1.57
EAFS-RF Model	1.76
BHPO-MLPAD	1.43
AOAEL-CDC	0.80

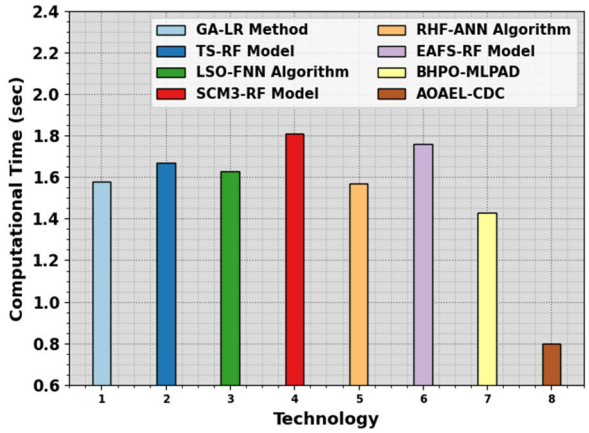

FIGURE 9. CT analysis of the AOAEL-CDC model with other algorithms.

In Table 4 and Fig. 9, the comparative computational time (CT) analysis of the AOAEL-CDC methodology is noticeably described. These simulation findings display that the EAFS-RF and SCM3-RF technique reveals increased performance. Concurrently, the LSO-FNN and TS-RF algorithms get moderately closed outcomes. Then, the GA-LR, RHF-ANN, and BHPO-MLPAD methods show better outcomes. However, the AOAEL-CDC system achieves higher performance with a minimum CT of 0.80s respectively.

Thus, the AOAEL-CDC technique can be utilized for enhanced cybersecurity in the IoT platform.

V. CONCLUSION

In this study, we have presented an AOAEL-CDC technique on the IoT environment. The presented AOAEL-CDC technique exploited the FS approach with an ensemble learning approach for cyberattack recognition and classification in an environment of IoT. In the developed AOAEL-CDC technique, the feature selection takes place by employing the AOA technique. For the cyberattack detection process, the ensemble learning process is carried out by the use of three models namely BiLSTM, GRU, and ELM. Finally, the hyperparameter selection of the DL models takes place utilizing MPA. To examine the performance analysis of the AOAEL-CDC technique, a series of simulations take place on the UNSW-NB15 dataset. The extensive simulation analysis concluded that the AOAEL-CDC technique reaches a higher accuracy of 99.31%

Future work in the realm of Cybersecurity in the IoT must focus on developing adaptive and context-aware security architectures. This involves the exploration of ML approaches like unsupervised and reinforcement learning, to optimize threat mitigation and anomaly detection abilities. Furthermore, there is a need for standardized security protocols that can be seamlessly operated into varied IoT devices, ensuring a robust and consistent security posture. The combination of blockchain technology to protect data exchanges and IoT transactions, along with the development of privacy-preserving systems, will be vital for addressing emerging problems.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2024-170-04".

REFERENCES

[1] M. F. Ansari, P. K. Sharma, and B. Dash, "Prevention of phishing attacks using AI-based cybersecurity awareness training," *Prevention*, vol. 3, no. 6, pp. 61–72, Mar. 2022.

[2] S. Subramanian, N. Venkatachalam, and R. Rajendran, "A novel phishing attack prediction model with crowdsouring in wireless networks," in *Perspectives on Social Welfare Applications' Optimization and Enhanced Computer Applications*. Hershey, PA, USA: IGI Global, 2023, pp. 31–51.

[3] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, Jan. 2021.

[4] A. A. Andryukhin, "Phishing attacks and preventions in blockchain based projects," in *Proc. Int. Conf. Eng. Technol. Comput. Sci. (EnT)*, Mar. 2019, pp. 15–19.

[5] Q. A. Al-Haija and S. Zein-Sabatto, "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks," *Electronics*, vol. 9, no. 12, p. 2152, Dec. 2020.

[6] M. Basheri and M. Ragab, "Quantum cat swarm optimization based clustering with intrusion detection technique for future Internet of Things environment," *Comput. Syst. Sci. Eng.*, vol. 46, no. 3, pp. 3784–3798, Sep. 2023.

[7] M. Elsisi, M.-Q. Tran, K. Mahmoud, D. A. Mansour, M. Lehtonen, and M. M. F. Darwish, "Towards secured online monitoring for digitalized GIS against cyber-attacks based on IoT and machine learning," *IEEE Access*, vol. 9, pp. 78415–78427, 2021.

[8] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Managing IoT cyber-security using programmable telemetry and machine learning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 60–74, Mar. 2020.

[9] M. Panda, A. A. A. Mousa, and A. E. Hassanien, "Developing an efficient feature engineering and machine learning model for detecting IoT-botnet cyber attacks," *IEEE Access*, vol. 9, pp. 91038–91052, 2021.

[10] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, and S. Hossain, "Phishing attacks detection using machine learning approach," in *Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Aug. 2020, pp. 1173–1179.

[11] O. A. Alkhudaydi, M. Krichen, and A. D. Alghamdi, "A deep learning methodology for predicting cybersecurity attacks on the Internet of Things," *Information*, vol. 14, no. 10, p. 550, Oct. 2023.

[12] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020.

[13] K. A. Alattas and A. Mardani, "A novel extended Internet of Things (IoT) cybersecurity protection based on adaptive deep learning prediction for industrial manufacturing applications," *Environ., Develop. Sustainability*, vol. 24, no. 7, pp. 9464–9480, Jul. 2022.

[14] L. A. Maghrabi, I. R. Alzahrani, D. Alsalman, Z. M. AlKubaisy, D. Hamed, and M. Ragab, "Golden jackal optimization with a deep learning-based cybersecurity solution in industrial Internet of Things systems," *Electronics*, vol. 12, no. 19, p. 4091, Sep. 2023.

[15] F. Farhin, I. Sultana, N. Islam, M. S. Kaiser, M. S. Rahman, and M. Mahmud, "Attack detection in Internet of Things using software defined network and fuzzy neural network," in *Proc. Joint 9th Int. Conf. Informat., Electron. Vis. (ICIEV) 4th Int. Conf. Imag., Vis. Pattern Recognit. (icIVPR)*, Aug. 2020, pp. 1–6.

[16] A. Bhattacharjee, "Cyber security intrusion detection deep learning model for Internet of Things (IoT)," Ph.D. dissertation, Nat. College Ireland, Dublin, Ireland, 2022.

[17] P. Vijayalakshmi and D. Karthika, "Hybrid dual-channel convolution neural network (DCCNN) with spider monkey optimization (SMO) for cyber security threats detection in Internet of Things," *Meas., Sensors*, vol. 27, Jun. 2023, Art. no. 100783.

[18] M. Tawfik, N. M. Al-Zidi, B. Alsellami, A. M. Al-Hejri, and S. Nimbhore, "Internet of Things-based middleware against cyber-attacks on smart homes using software-defined networking and deep learning," in *Proc. 2nd Int. Conf. Comput. Methods Sci. Technol. (ICCMST)*, Dec. 2021, pp. 7–13.

[19] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, 2023.

[20] M. Nanjappan, K. Pradeep, G. Natesan, A. Samydurai, and G. Premalatha, "DeepLG SecNet: Utilizing deep LSTM and GRU with secure network for enhanced intrusion detection in IoT environments," *Cluster Comput.*, pp. 1–13, Jan. 2024.

[21] H. Drias, Y. Drias, N. A. Houacine, L. S. Bendimerad, D. Zouache, and I. Khennak, "Quantum OPTICS and deep self-learning on swarm intelligence algorithms for COVID-19 emergency transportation," *Soft Comput.*, vol. 27, no. 18, pp. 13181–13200, Sep. 2023.

[22] M. Mafarja, T. Thaher, M. A. Al-Betar, J. Too, M. A. Awadallah, I. A. Doush, and H. Turabieh, "Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning," *Appl. Intell.*, vol. 53, no. 15, pp. 18715–18757, Aug. 2023.

[23] M. Ehtearm, H. Ghayoumi Zadeh, A. Seifi, A. Fayazi, and M. Dehghani, "Predicting hydropower production using deep learning CNN-ANN hybridized with Gaussian process regression and salp algorithm," *Water Resour. Manage.*, vol. 37, no. 9, pp. 3671–3697, Jul. 2023.

[24] A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe, P. Bhattacharya, B. Sharma, and S. Chowdhury, "Composition of hybrid deep learning model and feature optimization for intrusion detection system," *Sensors*, vol. 23, no. 2, p. 890, Jan. 2023.

[25] S. Das, T. P. Sahu, and R. R. Janghel, "Stock market forecasting using intrinsic time-scale decomposition in fusion with cluster based modified CSA optimized ELM," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8777–8793, Nov. 2022.

[26] A. T. Sahlol, D. Yousri, A. A. Ewees, M. A. A. Al-Qaness, R. Damasevicius, and M. A. Elaziz, "COVID-19 image classification using deep features and fractional-order marine predators algorithm," *Sci. Rep.*, vol. 10, no. 1, p. 15364, Sep. 2020.

[27] Accessed: Dec. 12, 2023. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[28] A. O. Khadidos, Z. M. AlKubaisy, A. O. Khadidos, K. H. Alyoubi, A. M. Alshareef, and M. Ragab, "Binary Hunter–Prey optimization with machine learning—Based cybersecurity solution on Internet of Things environment," *Sensors*, vol. 23, no. 16, p. 7207, Aug. 2023.