IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 December 2024, accepted 1 January 2025, date of publication 6 January 2025, date of current version 21 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3526258

RESEARCH ARTICLE

Boosting Cyberattack Detection Using Binary Metaheuristics With Deep Learning on Cyber-Physical System Environment

ALANOUD AL MAZROA[1], FAHAD R. ALBOGAMY[2], MOHAMAD KHAIRI ISHAK[3], (Member, IEEE), AND SAMIH M. MOSTAFA[4,5]

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2]Turabah University College, Computer Sciences Program, Taif University, Taif 21944, Saudi Arabia
[3]Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates
[4]Computer Science Department, Faculty of Computers and Information, South Valley University, Qena 83523, Egypt
[5]Faculty of Industry and Energy Technology, New Assiut Technological University (NATU), New Assiut City 71684, Egypt

Corresponding author: Fahad R. Albogamy (f.alhammdani@tu.edu.sa)

This work was supported by Taif University, Taif, Saudi Arabia, under Grant TU-DSPP-2024-129.

ABSTRACT The swift advancement of cyber-physical systems (CPSs) across sectors such as healthcare, transportation, critical infrastructure, and energy enhances the crucial requirement for robust cybersecurity measures to protect these systems from cyberattacks. The cyber-physical method is a hybrid of cyber and physical components, and a safety breach in the element is central to catastrophic consequences. Cyberattack recognition and mitigation techniques in CPSs include using numerous models like intrusion detection systems (IDSs), access control mechanisms, encryption, and firewalls. Cyberattack detection employing deep learning (DL) contains training neural networks to identify patterns indicative of malicious actions within system logs or network traffic, allowing positive classification and mitigation of cyber-attacks. By leveraging the integral ability of DL methods to learn complex representations, this technique enhances the accuracy and efficiency of detecting diverse and growing cyber-attacks. Thus, the study proposes an automated Cyberattack Detection using Binary Metaheuristics with Deep Learning (ACAD-BMDL) method in a CPS environment. The ACAD-BMDL method mainly focuses on enhancing security in the CPS environment via the cyberattack detection process. The ACAD-BMDL method uses Z-score normalization to scale the input dataset. In addition, the binary grey wolf optimizer (BGWO) model is utilized to choose an optimal feature subset. Moreover, the Enhanced Elman Spike Neural Network (EESNN) model detects cyber-attacks. Furthermore, the Archimedes Optimization Algorithm (AOA) model is employed to select the optimum hyperparameter for the EESNN model. The empirical analysis of the ACAD-BMDL technique is performed on a benchmark dataset. The experimental validation of the ACAD-BMDL technique portrayed a superior accuracy value of 99.12% and 99.36% under NSLKDD2015 and CICIDS2017 datasets in the CPS environment.

INDEX TERMS Cyber-physical systems, deep learning, cyberattack detection, metaheuristics, Archimedes optimization algorithm.

I. INTRODUCTION

The CPS is supervised and controlled by distributed sensors, embedded computers, and actuators linked via communication networks [1]. Today's modern life hugely depends on CPS owing to their vast applications in dissimilar regions like next-generation aerospace and transport systems, process control and water treatment systems, power methods, and smart grid [2]. Using CPS for these applications gives us unique abilities to achieve higher performance

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

and consistency in complex tasks. A CPS typically contains a network of interrelating units with physical gadgets and computational elements. The robust communication network requirement makes systems vulnerable to cyberattacks like deception and Denial of Service (DoS) attacks [3]. Those attacks are inserted into systems in the physical and cyber layers. Furthermore, only some malicious attackers concentrate on attacks between physical and cyber layers, which can significantly damage physical devices [4]. It must be noticeable that an attacker can randomly interrupt the system dynamics or prompt any perturbations to CPS without sufficient safety shields of software or hardware tactics, thus leading to significant general losses or the loss of human survival [5]. For example, the Iranian nuclear capability was hit by the Stuxnet malware, power blackouts in Brazil, blackout accidents in nuclear plants, etc. These instances specify an urgent necessity for trustworthy attack recognition schemes to contract with malicious attacks and preserve CPS performance. If cyberattacks are identified and placed quickly, the loss to complete systems is controlled within a supportable limit [6].

There are many tasks in this field, whereas recognizing the before unknown attacks is one of the most crucial. Another task links to the accessibility of the datasets employed to train logical methods, as the attack recognition performance is sturdily based on the excellence of the training dataset. The foremost task is connected to machine learning (ML) methods, which are generally trained on datasets with well-known attack designs. As an outcome, they cannot identify before hidden attacks [7]. The probable solution is to utilize anomaly recognition methods dependent upon studying the cyber-physical objects' behaviour. Anomaly-based methods analyze the standard network and behaviour of the system and recognize anomalies as abnormalities from the standard behaviour [8]. They are engaging owing to their capability to identify zero-day attacks. An additional benefit is that the profiles of everyday actions have been modified for each system, network, or application. Therefore, it will be more complex for attackers to recognize which actions they can implement unobserved. In addition, the data on anomaly-based models is employed to express the signature for wrong detectors [9]. The foremost drawback of anomaly-based models is the probability of a higher false alarm rate because before, behaviours of hidden systems could be classified as anomalies. Hybrid recognition integrates anomaly and misuse recognition. It is primarily utilized to reduce false positives for unknown attacks while enhancing detection rates for known intrusions. Most DL/ML approaches utilize hybrid approaches [10].

The study proposes an automated Cyberattack Detection using Binary Metaheuristics with Deep Learning (ACAD-BMDL) method in a CPS environment. The ACAD-BMDL method mainly focuses on enhancing security in the CPS environment via the cyberattack detection process. The ACAD-BMDL method uses Z-score normalization to scale the input dataset. In addition, the binary grey wolf optimizer (BGWO) model is utilized to choose an optimal feature subset. Moreover, the Enhanced Elman Spike Neural Network (EESNN) model detects cyber-attacks. Furthermore, the Archimedes Optimization Algorithm (AOA) model is employed to select the optimum hyperparameter for the EESNN model. The empirical analysis of the ACAD-BMDL technique is performed on a benchmark dataset. The significant contribution of the ACAD-BMDL method is as follows:

- The ACAD-BMDL model presents a novel approach to improving security in CPS through an advanced cyberattack detection process. It incorporates cyber and physical components to address a critical vulnerability. Moreover, the model integrates novel methods such as feature selection, neural network optimization, and hyperparameter tuning to enhance overall detection efficiency.
- The BGWO is employed to detect the optimal subset of features from the dataset. This technique confirms that only the most relevant factors are implemented for cyberattack recognition, improving the detection system's performance and effectiveness. Furthermore, this feature selection process mitigates computational complexity and enhances the detection model's overall accuracy.
- The ACAD-BMDL approach implements the EESNN model for cyberattack recognition, utilizing advanced neural network methods to improve accuracy and robustness. This contribution utilizes advanced neural network architectures to detect and respond to threats in CPS environments efficiently. The EESNN model also improves detection abilities by capturing complex patterns and temporal dependencies in the data.
- The ACAD-BMDL model utilizes the AOA model to determine the optimal hyperparameters for the EESNN technique. This contribution enhances model performance by systematically optimizing hyperparameters, enhancing detection accuracy and efficiency. Also, AOA's optimization process refines the technique's ability to adapt to complex cyberattack patterns in CPS environments.
- The study's novel aspect is its incorporated model. It integrates advanced optimization methods, such as Z-score normalization for scaling, BGWO for feature selection, EESNN for detection, and AOA for hyperparameter tuning. This holistic technique employs the strengths of every method to improve the security and efficiency of cyberattack detection in CPS environments, a unique and overall methodology not commonly seen in existing literature.

II. LITERATURE SURVEY

Alqahtani and Khan [11] developed an optimal hybrid transit search-cascade regional convolutional neural network (hybrid TS-Cascade R-CNN) for detecting cyberattacks. The model integrates a fusion TS model with a cascade regional convolutional network to improve detection accuracy by utilizing global and local indicators, efficiently detecting

and classifying advanced attacks. Chinaechetam et al. [12] presented an explainable deep neural network (DNN) by using the state-of-the-art cybersecurity datasets of IoT and an effectual feature selection (FS) model. This research paper delivers a visually interpretable, reliable, and measurable clarification of the Network IDS (NIDS) method utilizing Shapley Additive exPlanations (SHAP) and local interpretable model-agnostic explanation (LIME) explainability models. Dash et al. [13] offer a recognition strategy dependent upon the DL method. The recommended deep convolutional neural network (DCNN) detection method defines the traffic's network protocol before identifying Internet of Medical Things (IoMTs) defects. This research establishes how uneven data possibly disturbs the training procedure of the method and delivers an efficient tactic, the synthetic minority oversampling technique (SMOTE), to contract with such a situation. Chauhdary et al. [14] present a novel model that leverages evolutionary and DL techniques. The input data endures preprocessing to adapt it to an appropriate layout. In addition, Evolution Social Spider Optimization (ESSO) has been used. Then, the Deep Belief Network (DBM) model is trained using an Extreme Learning Machine (ELM). The performance of the developed system is enhanced over modification utilizing Poor and Rich Optimization (PRO) models.

Joshi et al. [15] developed an Improved Equilibrium Optimizer with an Enabled False Data Injection Attack Recognition (IEODL-FDIAR) model. In this method, the IEO technique is employed for the FS procedure. Besides, the presented model uses a stacked autoencoder (SAE) technique. Also, the pelican optimizer algorithm (POA) is exploited to select the hyperparameter of the SAE technique. Muthulakshmi et al. [16] present a new DL method for anomaly recognition in sensor-based CPS utilizing BiLSTM with Red Deer Algorithm (BiLSTM-RDA). At first, the sensor data experiences preprocessing. Next, the Bi-LSTM-based classification method takes place. Lastly, parameter alteration of the Bi-LSTM method has been executed by the custom of RDA for parameter tuning, for example, amount of unseen layers, learning rate, epoch count, and batch size. Sivamohan et al. [17] suggest a new IDS named TEA-EKHO-IDS that uses enhanced krill herd optimization (EKHO) and trustworthy explainable artificial intelligence (XAI) methods. The developed model utilizes XAI-EKHO for FS. The performance of intrusion detection is adjusted by combining Bayesian optimization and Bi-LSTM (BO-Bi-LSTM) for efficient recognition and identification. Kamble and Malemath [18] planned the IDS containing dual stages. The Adam Improved Rider Optimizer Approach (Adam IROA) has been proposed here. Here, the FS is completed utilizing the presented WWIROA model. This technique is projected by uniting IROA and WWO. The acquired features are supplied to the identification unit to determine the intrusions in the system. The classification is developed by applying Adam IROA-based DeepRNN. Nkoro et al. [12] propose an explainable DNN for recognizing network intrusions in Metaverse environments, utilizing SHAP and LIME for transparent, interpretable evaluation of network traffic from IoT devices, safeguarding precise detection of anomalies.

AlEisa et al. [19] present a DL-based IDS technique. An ensemble Long-Short Term Memory (LSTM) method is developed to detect malicious activities. Flavia and Chelliah [20] introduce a novel model for remote data integrity auditing and sanitizing, employing identity-based auditing with zero-knowledge proofs (ZK-SNARK and ZK-STARK). A Pinhole-Imaging-Based Learning Butterfly Optimization Algorithm with a Lightweight Convolutional Neural Network (PILBOA-LCNN) model is utilized to detect and protect sensitive terms in documents. In [21], a novel hybrid Autoencoder and Modified Particle Swarm Optimization (HAEMPSO) model is proposed for FS and DNN for classification. The PSO with modification of inertia weight was employed to optimize the DNN parameters. Alkahtani et al. [22] present the Election-Based Optimization Algorithm with DL for False Data Injection Attack Detection (EBODL-FDIAD) in CPS environments. It enhances security by detecting FDIAs using linear scaling normalization for data preprocessing and ensemble learning with kernel ELM (KELM), LSTM, and attention-based bidirectional recurrent neural network (ABiRNN) classifiers. The EBO model is employed for optimal hyperparameter selection. Chauhdary et al. [14] introduce a model integrating evolutionary and DL methods. The methodology employs ESSO for FS and a DBN trained with ELM classification, improving performance via PRO models. Antony and Revathy [23] propose the GO-3DANB methodology utilizing a three-stream, double attention network-modified BiLSTM and Gannet Optimization Algorithm (GOA) models to detect Sybils in OSNs by extracting key features.

III. THE PROPOSED MODEL

In this study, an ACAD-BMDL method is designed for the CPS platform. The technique mainly focuses on enhancing security in the CPS environment via the cyberattack detection process. The method involves different sub-processes: Z-score normalization, BGWO-based feature subset selection, EESNN-based classification, and AOA-based hyperparameter tuning. Fig. 1 depicts the entire flow of the ACAD-BMDL technique.

A. Z-SCORE NORMALIZATION

Initially, the ACAD-BMDL method underscores Z-score normalization, which is used for scaling the input dataset. Z-score normalization is a widely utilized model for scaling input datasets. It can standardize features by converting them into a common scale with a mean of zero and a standard deviation of one. This methodology confirms that every feature contributes equally to the method, reducing the influence of varying scales and units across varying features. The primary merit of Z-score normalization is its efficiency in enhancing the performance and convergence speed of many ML models by creating a uniform data dispersion.

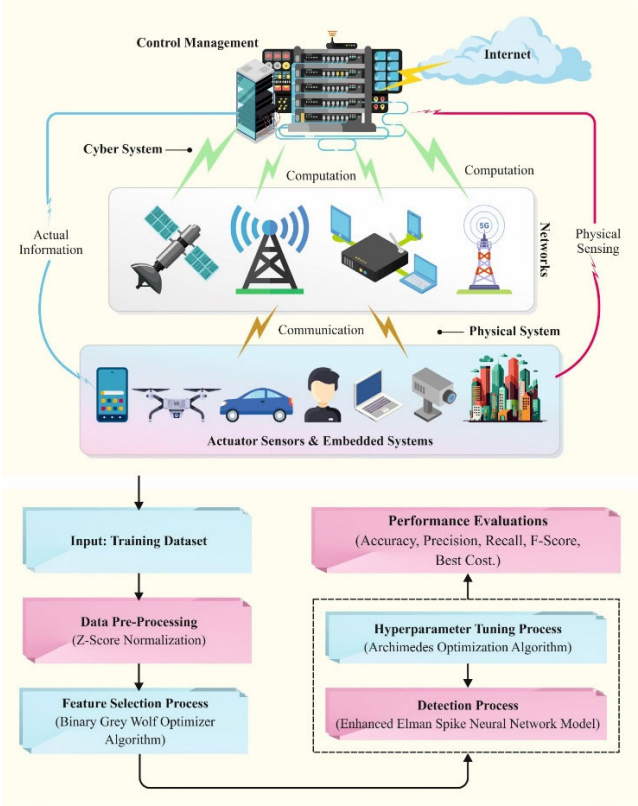

FIGURE 1. The overall flow of the ACAD-BMDL technique.

Unlike other scaling approaches, namely min-max normalization, which can compress outliers into a narrow range, Z-score normalization retains the relative data dispersion and is less sensitive to extreme values. This makes it specifically valuable in scenarios where data distributions vary widely and assists in attaining more stable and precise model training and evaluation.

Z-score normalization, also well-known as standardization, is a statistical model employed to change a dataset by rescaling its values to have a variance of 1 and a mean of 0. This procedure includes deducting the mean of the dataset from every data point and then separating the outcome by the SD. Z-score normalization is mainly beneficial when functioning with features with dissimilar scales, certifying that all variables donate similarly to examination, like clustering or ML methods. By standardizing the data, extreme values and outliers are efficiently measured in terms of their deviation from the mean, simplifying more substantial contrasts across variables.

Z-score normalization is utilized for scaling the input dataset, as it often improves the performance of techniques. However, to truly assess its impact, it is significant to quantify how normalization affects the algorithm's performance metrics.

To compute the normalization efficiency, evaluate the mean squared error (MSE) before and after applying Z-score normalization. The MSE is given by:

$$MSE = \frac{1}{n}\sum_{1}^{n}\left(y_i - \hat{y}_i\right)^2$$

where y_i depicts the actual values, and $\hat{y}_i$ specifies the anticipated values, by comparing the MSE values attained before and after normalization, you can quantitatively measure the impact of normalization on the model's accuracy.

A lower MSE after normalization would suggest that the normalization procedure enhanced the technique's performance, suggesting that scaling the input data reduced the prediction errors. On the contrary, if there is no crucial alteration or MSE increases, it may indicate that normalization did not benefit the technique or that other factors may influence performance.

B. BGWO-BASED FS

The BGWO method is applied to elect an optimum feature subset [24]. The BGWO model, drawing inspiration from the hunting tactics of grey wolves, is specifically efficient for FS due to its adept balance between exploration and exploitation. This technique's binary encoding aligns well with distinct FS tasks, giving a personalized methodology that improves effectiveness. The BGWO method's key advantage is its capacity to avert local optima by dynamically altering its search strategies. It is a significant merit over conventional approaches that may need help with high-dimensional or complex feature spaces. The capability of the BGWO technique to precisely detect and choose relevant features substantially enhances model performance. Furthermore, its robust optimization capabilities ensure a more precise and reliable feature subset, making it a preferred choice over another model in tackling complex FS issues. Fig. 2 illustrates the steps involved in the BGWO model.

Alpha (α), beta (β), delta (δ), and omega (ω) wolves are the four major types of grey wolves. Grey wolf lives in packs and follows social rules that rank hierarchically. Alpha wolves dominate the pack and make a decision. Beta assists the alpha in decision-making. Delta obeys alpha and beta; they lead omega wolves. Grey wolves try changing their location to reach their prey by following the three grey wolf packs above. Using this model, the wolf attempts to reach the target location in the best possible way. The BGWO models give a balance between exploration and exploitation in FS but are sensitive to parameter settings and convergence speed. Related to other methods, GA outperforms diversity, PSO is fast but may converge prematurely, ACO is flexible but computationally costly, SA can escape local optima but is slow, and wrapper methods provide model-specific benefits at a high computational cost.

In GWO, the wolf location refers to a candidate solution, whereas the prey location refers to the optimum solution. About the grey wolf solution, the best is α, the second

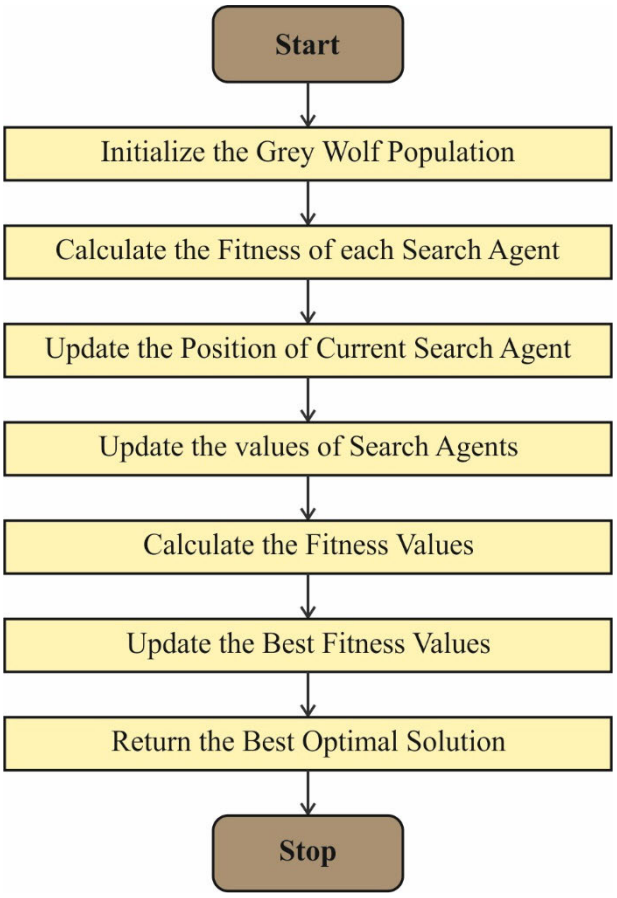

FIGURE 2. Steps involved in the BGWO model.

best is β, the third best is δ, and the rest is ω. The fitness value is evaluated after the wolf is randomly positioned. Next, each wolf should update the location relative to the top three wolves. Then, considering the above three wolves at each step, the model reiteratively updates the wolf position. The best wolf location and performance score, known as fitness value, are recorded in each iteration during this updating process. The primary objective is to locate the optimum location closer to the prey.

The best solutions at iteration t are denoted as $\vec{X}_\alpha, \vec{X}_\beta$, and $\vec{X}_\delta$. Eq. (1) determines $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$ as coefficient vectors.

$$\vec{A} = 2a \times \vec{r}_1 - a \tag{1}$$

where $\vec{r}_1$ is a random vector produced within [0, 1]. The distance control factor balances the tradeoff between exploitation and exploration, starting with a high value equivalent to 2 and dropping linearly until it attains 0. t and $t_{\max}$ are the existing iteration and the overall iteration counter. The factor a is evaluated by Eq. (2),

$$a = 2 - t \times \frac{2}{t_{\max}} (2) \tag{2}$$

Eqs. (3), (4), & (5) define $\vec{D}_\alpha, \vec{D}_\beta$, and $\vec{D}_\delta$ as distance vectors between α, β, δ and i.

$$\vec{D}_\alpha = \left| \vec{C}_1 \times \vec{X}_\alpha - \vec{X}_i \right| \tag{3}$$

$$\vec{D}_\beta = \left| \vec{C}_2 \times \vec{X}_\beta - \vec{X}_i \right| \tag{4}$$

$$\vec{D}_\delta = \left| \vec{C}_3 \times \vec{X}_\delta - \vec{X}_i \right| \tag{5}$$

Eq. (6) determines coefficient vectors as $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$; vector $\vec{r}_2$ is a randomly generated number ranging from zero to one. The error analysis for the Eqs. (3)-(5) comprises computing how measurement and computational inaccuracies in the vectors $\vec{C}_2$, $\vec{C}_3$, $\vec{X}_\beta$, $\vec{X}_\delta$, and $\vec{X}_i$ propagate through the cross-product, subtraction, and magnitude calculations, affecting the outcomes. Quantitative analysis of error bounds and sensitivity assists in determining the impact on $\vec{D}_\beta$ and $\vec{D}_\delta$.

$$\vec{C} = 2 \times \vec{r}_2 \tag{6}$$

The BGWO detects them in the hypercube search range within [0, 1], while the GWO locates the wolf position in continuous space. Set some equations for updating the wolf position to move further away from them or closer to the hypercube.

The continuous step size value, s_1^d, s_2^d, and $s_3^{d,is\ computed}$ using the sigmoid function. Now, d indicates the dimension of the search range.

$$s_1^d = \frac{1}{1 + \exp\left(-10\left(A^d \times D_\alpha^d - 0.5\right)\right)} \tag{7}$$

$$s_2^d = \frac{1}{1 + \exp\left(-10\left(A^d \times D_\beta^d - 0.5\right)\right)} \tag{8}$$

$$s_3^d = \frac{1}{1 + \exp\left(-10\left(A^d \times D_\delta^d - 0.5\right)\right)} \tag{9}$$

where $r3$ is a random value ranging from zero to one.

$$bstep_1^d = \begin{cases} 1 & if\ s_1^d \geq r3 \\ 0 & else \end{cases} \tag{10}$$

$$bstep_2^d = \begin{cases} 1 & if\ s_2^d \geq r3 \\ 0 & else \end{cases} \tag{11}$$

$$bstep_3^d = \begin{cases} 1 & if\ s_3^d \geq r3 \\ 0 & else \end{cases} \tag{12}$$

The sigmoid functions s_1^d, s_2^d, and s_3^d utilize a parameter A^d to control the steepness of the transition between FS states. A higher A^d makes the sigmoid function steeper, leading to more abrupt FS changes and potentially faster but less stable convergence. On the contrary, a lower A^d results in a gentler sigmoid curve, promoting gradual FS changes and more stable but slower convergence. The choice of A^d thus affects both the rate and reliability of convergence, with steeper curves causing quicker and more decisive selections. In comparison, gentler curves improve exploration but may need more iterations to attain optimal outcomes.

Furthermore, compared to other thresholding techniques, the binary conversion using $bstep^d$ (based on sigmoid outputs s_1^d, s_2^d, s_3^d and a random threshold $r3$) often gives a more complex and adaptive selection of features, dynamically altering to feature significance and distribution, which can enhance optimization performance and robustness.

The distance i move towards α, β and δ is represented by $bstep_1$, $bstep_2$, and $bstep_3$.X_1, X_2, and X_3 correspondingly indicate the binary vectors affecting the movement of α, β, and δ wolves.

$$X_1^d = \begin{cases} 1 & if\ X_\alpha^d + bstep_1^d \geq 1 \\ 0 & else \end{cases} \tag{13}$$

$$X_2^d = \begin{cases} 1 & if\ X_\beta^d + bstep_2^d \geq 1 \\ 0 & else \end{cases} \tag{14}$$

$$X_3^d = \begin{cases} 1 & if\ X_\delta^d + bstep_3^d \geq 1 \\ 0 & else \end{cases} \tag{15}$$

After attaining X_1^d, X_2^d, and X_3^d, the location of X_i at the $(t+1)$ iteration is updated by stochastic crossover. $r4$ denotes a random integer selected from the standard distribution $\in [O, 1]$

$$X_i^d\ (t+1) = \begin{cases} X_1^d & if\ r4 < \frac{1}{3} \\ X_2^d & elseif\ \frac{1}{3} \leq r4 < \frac{2}{3} \\ X_3^d & else \end{cases} \tag{16}$$

The FF considers the number of selected features and the classifier's performance. It reduces the set size of selected attributes and improves the classifier's accuracy. Thus, the FF is used to estimate individual solutions.

$$Fitness = \alpha * ErrorRate + (1-\alpha) * \frac{\#SF}{\#All\_F} \tag{17}$$

In Eq. (17), *ErrorRate* implies the classifier errors using the attributes selected. ErrorRate is the percentage of incorrect classification to the number of classifications made within [0, 1]. $\#SF$ indicates the amount of attributes chosen, and $\#All\_F$ indicates the overall quantity of features in the original dataset. The parameter α controls the classification quality importance and subset length, whose value is set to 0.9.

The FS process involves detecting and choosing the most relevant attributes from a dataset to improve the model's performance. The BGWO technique plays a significant role by implementing binary encoding for efficient feature selection. BGWO indulgences candidate feature subsets as wolves, with the optimal subset portrayed by the prey. Through iterative updates of wolf positions relative to the top three wolves (α, β, δ), based on their fitness values, BGWO explores and exploits the feature space to find the best feature subset. This process utilizes sigmoid functions and binary thresholds to adjust FS dynamically. Usually, BGWO results in an enhanced performance and a reduced feature set. For example, it may choose substantial features such as "network traffic volume," "protocol types," and "packet sizes" to detect cyberattacks, improving both accuracy and efficiency. The capability of the BGWO model to balance exploration and exploitation allows it to avoid local optima and converge on a feature set that improves model performance while reducing complexity.

C. CYBERATTACK DETECTION USING EESNN MODEL

The EESNN model is exploited to detect cyber-attacks [25]. The EESNN method is highly efficient for cyberattack detection due to its advanced architecture incorporating spike-based neural processing with the Elman network's feedback mechanisms. This approach outperforms capturing temporal dependencies and dynamic patterns in cybersecurity data, which are substantial enough to detect complex attack behaviours and anomalies. The EESNN model's advantage is its capability to process and learn from spatiotemporal sequences of network activity, improving its sensitivity to subtle and evolving attack signatures. Unlike conventional neural networks, EESNN benefits from its spike-timing dependent plasticity, which enhances learning effectiveness and robustness in detecting advanced threats. Moreover, its improved feedback loops allow for better adaptation and real-time detection, making EESNN a greater choice than conventional methods that may face difficulty with the dynamic and growing nature of cyberthreats. Fig. 3 demonstrates the infrastructure of EESNN.

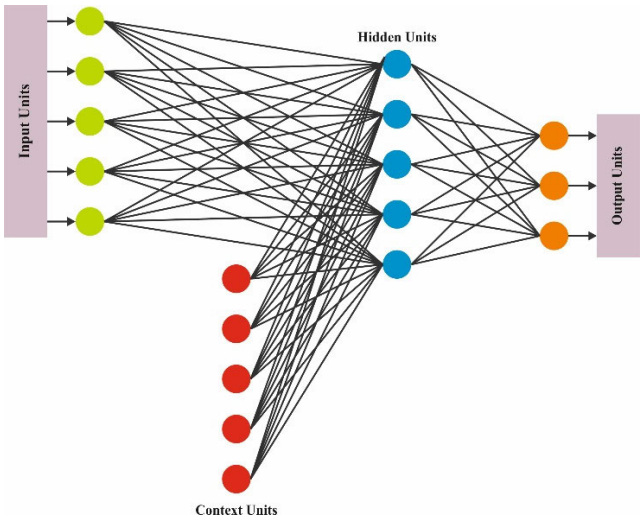

FIGURE 3. Structure of EESNN.

After the feature extraction process, the extracted features are fed into the EESNN classification. Using elementary ENN, this kind of partial recurrent SNN model adjusts the EESNN. The typically made topology of ENN is the hidden layer (HL), context layer, input layer, and output layer. The context layer stores the previous output of HL using a positive feedback process. Due to its smaller weight, the EESN is ideally suited for calculating thorough application in edge devices and embedded systems. The nodes and input layer

are represented as follows:

$$y_i^{(1)}(m) = f_i^{(1)}\left(net_i^{(1)}(m)\right); i = 1 \tag{18}$$

Consider $net_i^{(1)}(m) = e_i^{(1)}(m)$: n denotes the n^{th} iterations, $e_i^{(1)}(m)$ as input, $y_i^{(1)}(m)$ shows the initial layer outcome. The dynamics of EESNN are labelled in Eqs. (19)-(21),

$$NS(y) = NLF\left(W_{con*inv}NS_{con}(y), W_{in*inv}input(y)\right) \tag{19}$$

$$NS_{Conv}(y) = \alpha(y)\, NS_{con}(y-1) + W * NS(y-1) \tag{20}$$

$$output(y+1) = W_{inv*out}NS(y) \tag{21}$$

$NLF(\cdot)$ is a nonlinear function that defines how EESNN is presented and categorized. The *output* (y) indicates the input and output, respectively. $NS(y)$ and $NSCon(y)$ represent the state node vector for the HL and context layers correspondingly. "w_{in*inv}" and "$w_{con*inv}$" are neurons connecting the weight to the HL, and "$w''_{inv*out}$ denotes the neurons weighted the HL to out.

$$y_j^{(2)}(n) = S\left(net_j^{(2)}(n)\right); \quad j = 1, \ldots 9 \tag{22}$$

$$net_j^{(2)}(n) = \sum\nolimits_i W_{in*inv} \times y_i^{(1)}(n) + \sum\nolimits_k W_{con*inv} \times y_k^{(3)}(n); \quad k = 1 \ldots .9 \tag{23}$$

where $S(net_j^{(2)}(n))$ represents the sigmoid function, $y_i^{(1)}(n)$, $y_k^{(3)}(n)$ denotes the data from the HL and input layers, and $y_j^{(2)}(n)$ signifies the HL outcome. α is self-connection feedback gain where the context layer is updated to achieve accurate malware classification.

$$y_k^{(3)}(n) = \alpha y_k^{(3)}(n-1) + y_j^{(2)}(n-1) \tag{24}$$

Weight and delay are explicit to sub-connection. In the EESNN, the layer-to-layer connection has the equivalent amount of synaptic terminals. 2 neurons provide output and two input neurons in the single-layer EESNN that make up HL.

$$y_l^{(4)}(n) = f_l^{(4)}\left(net_l^{(4)}(n)\right) \tag{25}$$

In Eq. (25), $f_l^{(4)}$ is the parameter controlled by the proposed method,

$$net_l^{(4)}(n) = \sum\nolimits_j W_{inv*0ut} \times y_j^{(2)}(n) \tag{26}$$

In Eq. (26), $w_{inv*out}$ indicates the neural weight that associates the layer viz., HL output to neuron.

$$W_{inv*out}^{y}(Time+1) = W_{inv*out}^{y}(Time) - \eta.\delta_f.NS^{y} \tag{27}$$

In Eq. (27), '*Time*' represents the unit of time, and η shows the learning rate. The threshold values of neurons inspect the IDS classification in all the neurons at the *Time*. $W_{inv*out}^{y}(Time+1)$ denotes the weight update at the next time step (Time + 1). $W_{inv*out}^{y}(Time)$ specifies the weight at the current time step (Time). η is the learning rate, a standard parameter in optimization algorithms that controls the size of the update step. δ_f depicts the error term or the gradient of the loss function concerning the weights. NS^{y} denotes the normalization or scaling factor that alters the update magnitude.

$$\delta_f = \frac{Error}{\sum_{i=1}^{Niv}\sum_{y=1}^{NoD} W_{inv*out}^{y}\frac{\partial input}{\partial Time}} \tag{28}$$

$$Error = t_f^{DST} - Time_f^{NLF} \tag{29}$$

where t_f^{DST} specifies the neuron's spike duration, $Time_f^{NLF}$ shows the actual spike timing of neuron outcome.

D. HYPERPARAMETER TUNING PROCESS

Lastly, the AOA is utilized to select the optimum hyperparameter for the EESNN method [26]. The AOA model is an ideal choice for hyperparameter tuning due to its innovative approach motivated by the principles of Archimedean mechanics. This approach outperforms in navigating the hyperparameter space with a robust exploration-exploitation balance, driven by its unique spiral search mechanism that efficiently converges on optimal solutions. The AOA technique's primary advantage is its ability to escape local optima and handle intrinsic, high-dimensional hyperparameter landscapes related to conventional optimization techniques with greater effectualness. Its adaptive nature allows it to dynamically alter search strategies, improving its flexibility and precision in tuning hyperparameters. Furthermore, the systematic approach of the AOA model gives a structured yet versatile framework for exploring diverse hyperparameter settings, resulting in enhanced model performance and stability. This makes it a greater alternative to other methods that may lack the same level of effectiveness and adaptability in hyperparameter optimization. Fig. 4 portrays the flow of the AOA technique.

AOA is a physics-based technique inspired by Archimedes' principles. The AOA model is chosen for hyperparameter tuning because it can effectively explore complex search spaces and optimize various parameters concurrently. Its robust and adaptive nature enables it to find optimal hyperparameters more efficiently than conventional techniques, enhancing model performance and precision.

Consequently, an object submerged in a liquid at the equilibrium state where the buoyancy force and the object's weight are equivalent is pushed up by the buoyant force comparable to the weight of the liquid displaced.

$$F_b = w_o; p_b v_b a_b = p_o v_o a_o \tag{30}$$

In Eq. (30), p, v, and a are the density, volume, and acceleration. b denotes the fluid, and 0 is applied for the immersed objects. The weight and volume of the object define the movement speed in the liquid. Submerged objects form a population in the AOA. Like other optimization techniques, the algorithm randomly searches for the solution. The density and volume of the objects are updated throughout iteration until the termination condition is reached. The steps of AOA are given below:

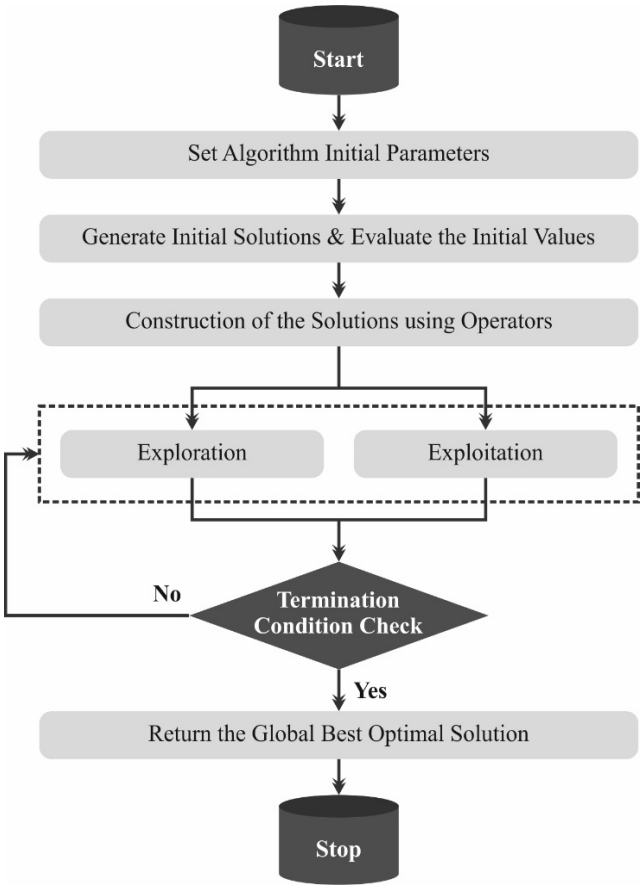

FIGURE 4. The workflow of the AOA model.

Step 1: the object values are randomly allocated as follows:

$$O_i = lb_i + rand \times (ub_i - lb_i)\, i = 1, 2, \cdots, N \quad (31)$$

Let o_i be the i^{th} objects in a population of N objects. lb_i and ub_istand for the lower and upper boundaries of the search region. Volume (vol) and density (den) are initialized randomly, and acceleration is initialized using the lower and upper boundary values.

$$den_i = rand, vol_i = rand \quad (32)$$

$$acc_i = lb_i + rand \times (ub_i - lb_i) \quad (33)$$

Step 2: density and volume values are updated using the following equation.

$$\begin{aligned} den_i^{t+1} &= den_i^t + rand \times \left(den_{best} - den_i^t\right); vol_i^{t+1} \\ &= vol_i^t + rand \times \left(vol_{best} - vol_i^t\right) \end{aligned} \quad (34)$$

Now, the optimal values of volume and density are vol_{best} and den_{best}.

Step 3: The density factor is decreased, and the transfer operator is increased to assist in the shift from exploration to exploitation phases after the collision of the object.

$$TF = \exp\left(\frac{t - t_{\max}}{t_{\max}}\right) \quad (35)$$

In Eq. (35), $tmax$ refers to the maximal iterations, and t indicates the iteration count. d indicates the density-reducing factor, and it reduces over time:

$$d^{t+1} = \exp\left(\frac{t_{\max} - t}{t_{\max}}\right) - \left(\frac{t}{t_{\max}}\right) \quad (36)$$

Step 4: Exploration stage: The collision occurs according to the TF values.

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \quad (37)$$

In Eq. (37), den_i, vol_i, and acc_i are the density, volume, and acceleration of object i. mr denotes the corresponding value of random material.

Step 5: Exploitation stage: here, the collision does not occur according to the TF value.

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \quad (38)$$

In Eq. (38), acc_{best} denotes the optimum acceleration values.

Step 6. Normalize acceleration: Here, the acceleration values are higher where they are away from the global minima, and they reduce over time.

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{\max(acc) - \min(acc)} + l \quad (39)$$

In Eq. (39), u and 1 are normalization's upper and lower boundaries. $acc_{i,nonn}^{t+1}$ controls the change of stepsize of all the objects.

Step 7: Update step. Here, the new position is updated and calculated.

When $TF \leq 0.5$, the exploration stage occurs.

$$x_i^{t+1} = x_i^t + C_1 \times rand \times acc_{i-norm}^{t+1} \times d \times \left(x_{rand} - x_i^t\right) \quad (40)$$

In Eq. (40), C_1 is equivalent to 2. If $TF > 0.5$, exploitation stages take place, and object position is updated by Eq. (41). The random values and constants impact convergence stability and effectualness by influencing the extent and direction of exploration. The parameter C_1 controls the scaling of the update step: with $C_1 = 2$, it balances exploration and exploitation, confirming efficient search while maintaining stability. Lower values such as $C_1 = 1.5$ tend to favour stability but can slow convergence, while higher values such as $C_1 = 2.5$ improve exploration and speed up convergence but risk instability and premature convergence. Thus, finding an optimal C_1 is significant for achieving efficient exploration and stable convergence.

$$\begin{aligned} x_i^{t+1} &= x_{best} \times F \times C_2 \times rand \times acc_{i-norm}^{t+1} \\ &\quad \times d \times \left(T \times xbest - x_i^t\right) \end{aligned} \quad (41)$$

where $C_2 = 6.T = C_3 \times TFT$ increases over time within $[C_3 \times 0.3, 1]$ .F refers to the fag used to change the direction:

$$F = \begin{cases} +1 & if\ P < 0.5 \\ -1 & if\ P > 0.5 \end{cases} \quad (42)$$

Step 8: Evaluation step. Here, the FF is estimated, and the result is compared. If the best results are found, the position is remembered.

Fitness selection is the critical factor that affects AOA performance. The hyperparameter selection technique includes the solution encoding method to estimate the accuracy of the solution candidate. Now, the AOA method considers performance the primary criterion for designing the FF.

$$Fitness = \max(P) \tag{43}$$

$$P = \frac{TP}{TP + FP} \tag{44}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{45}$$

$$Recall = \frac{TP}{TP + FN} \tag{46}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{47}$$

where the Eqs. (43)-(47) calculate fitness, precision-based fitness, F1-Score, precision, recall, and MCC.

In the equation, TP and FP are the true and the false positive values.

IV. RESULT ANALYSIS AND DISCUSSION

The experimental result analysis of the ACAD-BMDL technique is provided in this section. The ACAD-BMDL technique has chosen 12 features out of 41 from the NSLKDD2015 [27] dataset and 14 features out of 80 from the CICIDS2017 [28] dataset. The suggested technique is simulated using the Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are provided: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

Table 1 and Fig. 5 demonstrate the best cost (BC) outcomes of the ACAD-BMDL method with the applied datasets. On the NSLKDD2015 dataset, the ACAD-BMDL technique offers the least BC of 0.031. At the same time, the Stochastic Clonal Algorithm for Variable-Order FS (SCAVO-FS), Adaptive Harmony Search Algorithm for FS (AHSA-FS), Binary Beetle Antennae Search Algorithm for FS (BBFO-FS), Binary Firefly Optimization for FS (BFO-FS), Social Spider Optimization for FS (SSO-FS), and Whale Optimization Algorithm for FS (WOA-FS) models have reported improved BC of 0.053, 0.057, 0.076, 0.097, 0.106, and 0.122, correspondingly. Also, based on the CICIDS2017 dataset, the ACADBMDL technique gets a reduced BC of 0.036; however, the SCAVO-FS, AHSA-FS, BBFO-FS, BFO-FS, SSO-FS, and WOA-FS models stated higher BC of 0.041, 0.044, 0.067, 0.089, 0.096, and 0.120.

The confusion matrices of the ACAD-BMDL technique under NSLKDD2015 and CICIDS2017 datasets are shown

TABLE 1. BC outcome of the ACAD-BMDL model under NSLKDD2015 and CICIDS2017 datasets.

BC		
Methods	**NSLKDD2015**	**CICIDS2017**
ACADBMDL	0.031	0.036
SCAVO-FS	0.053	0.041
AHSA-FS	0.057	0.044
BBFO-FS	0.076	0.067
BFO-FS	0.097	0.089
SSO-FS	0.106	0.096
WOA-FS	0.122	0.120

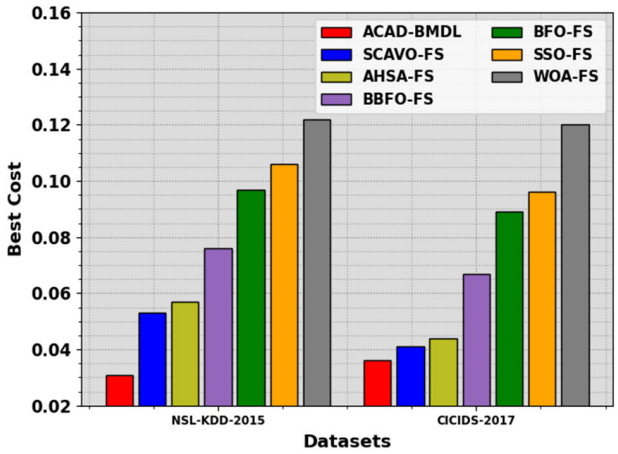

FIGURE 5. BC outcome of the ACAD-BMDL technique under two datasets.

in Fig. 6. In the training phase of the NSL-KDD-2015 dataset, the model precisely identified 46,811 anomalies and 40946 normal instances, with a low misclassification rate. In the testing phase, the model achieved a similar accuracy, with 20,189 anomalies and 17,385 normal instances anticipated correctly. For the CICIDS-2017 dataset, the training and testing phases exhibited the model's consistent performance, identifying 34,623 anomalies, 34,844 normal instances in training, 14,940 anomalies, and 14,838 normal instances in testing.

The comparison results of the ACAD-BMDL technique with the NSL-KDD dataset are given in Table 2 and Fig 7. These experimentation outcomes imply the ineffectual performance of the Gated Recurrent Unit (GRU) model. Next to that, the SCAVO with Enhanced Adaptive Evolutionary Intrusion Detection (SCAVO-EAEID), Probabilistic Robust Optimal DL-Based Intrusion Detection for CPS (PRO-DLBIDCPS), BBFO-GRU, and optimal GRU methods have exhibited closer performance. However, the ACAD-BMDL technique gains maximum training $accu_y$ values of 99.12%, $prec_n$ of 99.67%, $reca_l$ of 99.30%, and F_{score} of 99.93%.

The classified outcome of the ACAD-BMDL technique with the NSL-KDD dataset is graphically illustrated in Fig. 8 using training accuracy (TRAA) and validation accuracy (VALA) curves. This result exhibits a valuable analysis

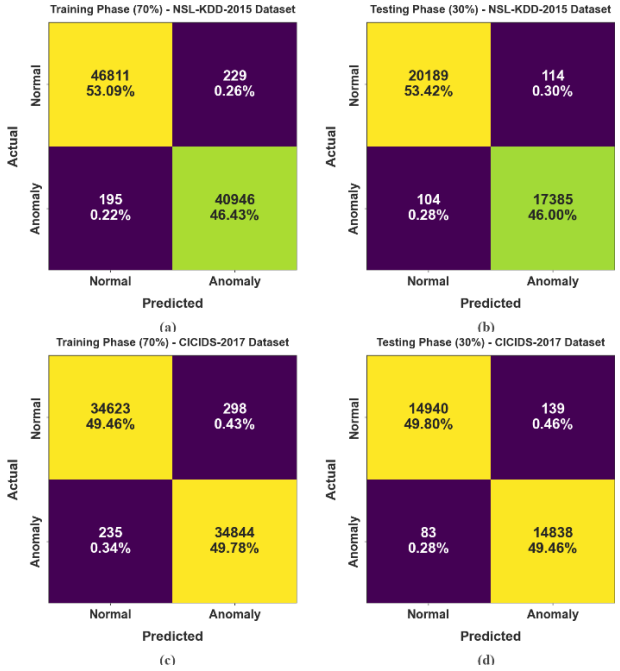

FIGURE 6. Confusion matrices of the ACAD-BMDL model under NSLKDD2015 and CICIDS2017 datasets (70:30).

TABLE 2. Comparison outcomes of the ACAD-BMDL method under the NSLKDD2015 dataset.

NSL-KDD Dataset				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
ACAD-BMDL	99.12	99.67	99.30	99.93
SCAVO-EAEID	98.96	99.51	99.13	99.83
PRO-DLBIDCPS	98.62	99.16	98.83	99.60
BBFO-GRU	98.34	98.95	98.47	99.21
Optimal GRU	98.04	98.45	98.00	98.70
GRU	97.70	98.17	97.63	98.31

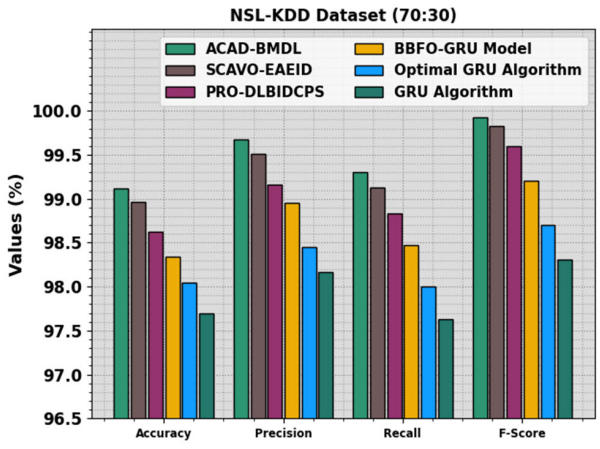

FIGURE 7. Comparison outcome of the ACAD-BMDL model with the NSLKDD2015 dataset (70:30).

of the behaviour of the ACAD-BMDL technique over varying epoch counts, establishing its learning model and generalization capabilities. Noticeably, the figure assumes a constant improvement in the TRAA and VALA with increasing epoch count. It ensures that the adaptive form of the ACAD-BMDL approach with a pattern recognition process is used for both data. The increased trends in VALA outline the capability of the ACAD-BMDL method to modify the TRA data and surpass exact classification on unnoticed data, pointing out the robust generalization abilities.

Fig. 9 shows an extensive review of the training loss (TRLA) and validation loss (VALL) results of the ACAD-BMDL technique with the NSL-KDD dataset over multiple epochs. The progressive decrease in TRLA indicates the ACAD-BMDL technique improving the weights and minimizing the classification error on both datasets. The figure specifies a clear understanding of the ACAD-BMDL technique related to the TRA dataset, emphasizing its ability to capture patterns within both datasets. Notably, the ACAD-BMDL technique incessantly improves its parameters in reducing the variance between the prediction and real TRA classes.

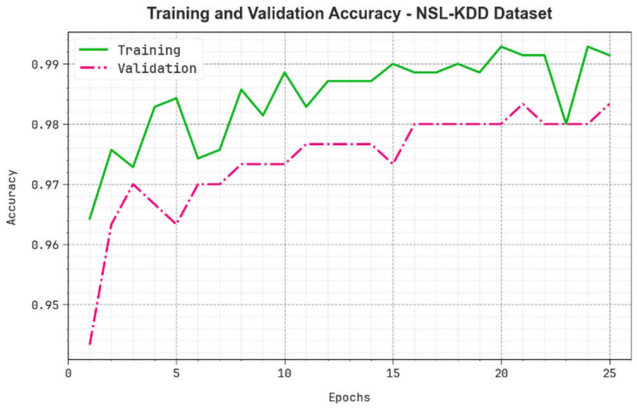

FIGURE 8. $Accu_y$ curve of the ACAD-BMDL model with the NSLKDD2015 dataset.

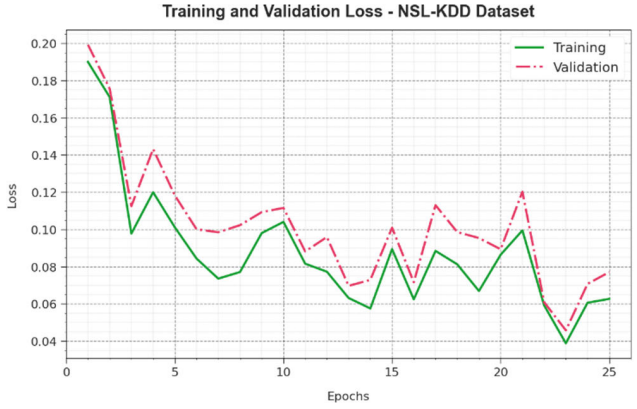

FIGURE 9. Loss curve of the ACAD-BMDL model with the NSLKDD2015 dataset.

Figs. 10 and 11 illustrate the convergence curve of the ACAD-BMDL technique under NSL-KDD and

CICIDS2017 datasets. The figures portrayed superior performance using convergence compared to other optimization models, namely the Flower Optimization Algorithm (FOA), Salp Swarm Algorithm (SCA), Moth-Flame Optimization (MFO), Arithmetic Optimization Algorithm (AOA), and Particle Swarm Optimization (PSO). The convergence graph clearly emphasizes that the ACAD-BMDL model converges faster and achieves more accurate outcomes, surpassing these established methods in efficiency and effectiveness across diverse benchmark tests.

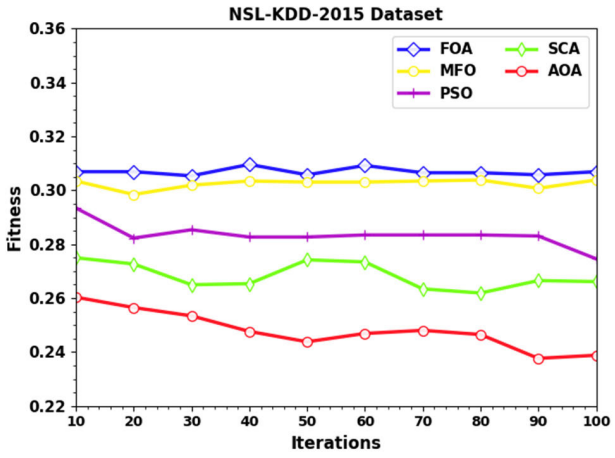

FIGURE 10. Convergence curve of the ACAD-BMDL model with the NSLKDD2015 dataset.

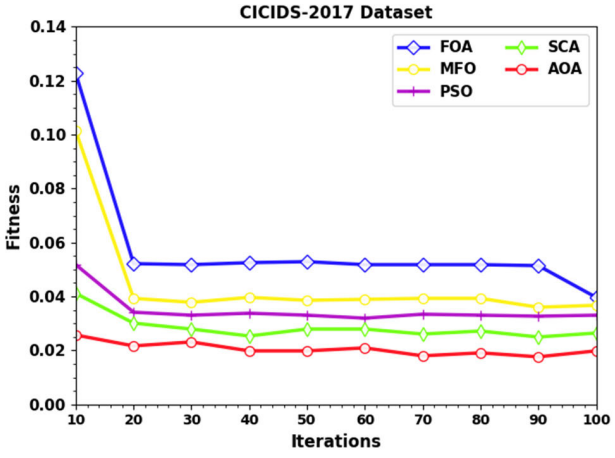

FIGURE 11. Convergence curve of the ACAD-BMDL model with the CICIDS2017 dataset.

Table 3 and Fig. 12 describe an extensive comparative result of the ACAD-BMDL method under the CICIDS2017 dataset. These outcomes indicate the poorer performance of the GRU method. Meanwhile, the SCAVO-EAEID, PRO-DLBIDCPS, BBFO-GRU, and optimal GRU models present closer performance. Nevertheless, the ACAD-BMDL models attain higher training $accu_y$ values of 99.36%, $prec_n$ of 99.69%, $reca_l$ of 99.58%, and F_{score} of 99.80%, appropriately.

TABLE 3. Comparison outcomes of the ACAD-BMDL model under the CICIDS2017 dataset.

CICIDS2017 Dataset				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
ACAD-BMDL	99.36	99.69	99.58	99.80
SCAVO-EAEID	99.18	99.54	99.42	99.62
PRO-DLBIDCPS	98.83	99.27	99.14	99.21
BBFO-GRU	98.51	98.93	98.70	98.72
Optimal GRU	98.07	98.71	98.33	98.42
GRU	97.81	98.36	98.07	98.03

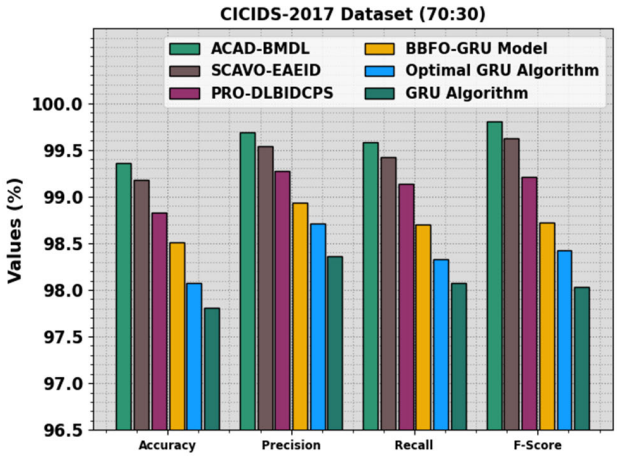

FIGURE 12. Comparison outcome of the ACAD-BMDL technique under the CICIDS2017 dataset (70:30).

The classifier results of the ACAD-BMDL technique with the CICIDS2017 dataset are displayed in Fig. 13 in the form of TRAA and VALA curves. The figure infers a valuable analysis of the behaviour of the ACAD-BMDL technique over several epochs, signifying its learning process and generalization abilities. Prominently, the figure considers a continual progression in the TRAA and VALA with progress in epochs. It ensures the adaptive form of the ACAD-BMDL method with a pattern recognition model for the TRA and TES data. The higher trends in VALA outline the ability of the ACAD-BMDL method to adapt to the TRA data and also surpass in giving an exact classifier on unnoticed data, indicating robust generalization abilities.

Fig. 14 demonstrated a wide-ranging analysis of the TRLA and VALL results of the ACAD-BMDL method with the CICIDS2017 dataset over diverse epochs. The progressive decrease in TRLA indicates that the ACAD-BMDL method improves the weights and decreases the classification error

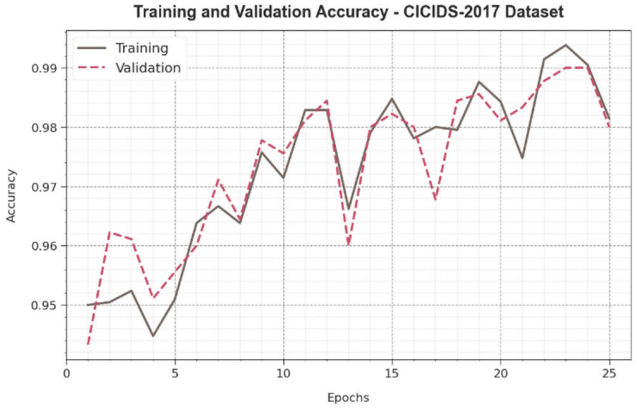

FIGURE 13. *Accu*$_y$ curve of the ACAD-BMDL technique under the CICIDS2017 dataset.

under both datasets. The figure identifies the clear understanding of the ACAD-BMDL technique correlated with the TRA data, emphasizing its ability to capture patterns within both datasets. Notably, the ACAD-BMDL method constantly improves its parameters to minimize the variances between the prediction and real TRA classes.

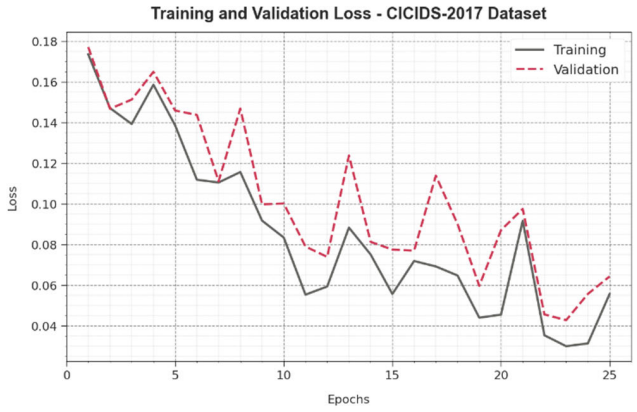

FIGURE 14. Loss curve of the ACAD-BMDL technique with the CICIDS2017 database.

The overall comparative assessment of the ACAD-BMDL method is given in Table 4 and Fig. 15 [29], [30]. These findings indicate that the MLIDS model performs worst, whereas the DT, GA-Fuzzy, and Fuzzy C-Means (FCM) models have illustrated moderately improved results. Next, the Cuckoo Optimization (CO), BBFO-GRU, SCAVO-EAEID, and PRO-DLBIDCPS models have demonstrated considerable performance over others. However, the ACAD-BMDL technique gains superior performance with a maximum $accu_y$ of 99.36%.

The training time (TRAT) and testing time (TEST) comparative outcomes of the ACAD-BMDL technique are determined and shown in Table 5 and Fig. 16. These accomplished outcomes denote that the FCM model obtains the higher performance, although the DT, GA-Fuzzy, and MLIDS techniques have shown moderately

TABLE 4. *Accu*$_y$ analysis of the ACAD-BMDL model with other techniques under the CICIDS2017 dataset.

Methods	$Accu_y$ (%)
ACAD-BMDL	**99.36**
SCAVO-EAEID	99.18
PRO-DLBIDCPS	98.83
BBFO-GRU	98.51
Decision Tree	96.87
MLIDS	94.03
CO	98.49
GA-Fuzzy	97.53
FCM	97.42

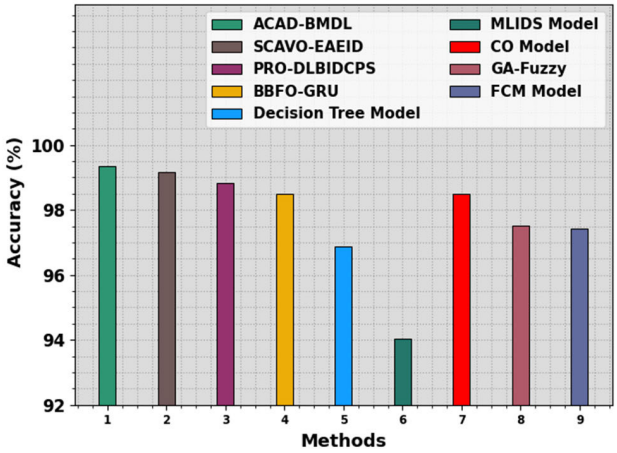

FIGURE 15. *Accu*$_y$ result of the ACAD-BMDL technique with other models.

TABLE 5. TRAT and TEST outcomes of the ACAD-BMDL model compared with other techniques.

Methods	Training Time (min)	Testing Time (min)
ACAD-BMDL	0.30	0.10
SCAVO-EAEID	0.59	0.29
PRO-DLBIDCPS	0.77	0.42
BBFO-GRU	1.13	0.41
Decision Tree	0.91	0.70
MLIDS	1.28	0.45
CO	0.85	0.61
GA-Fuzzy	1.39	0.49
FCM	1.80	0.90

boosted results. Similarly, the CO, BBFO-GRU, SCAVO-EAEID, and PRO-DLBIDCPS models have indicated significant performance over others. However, the ACAD-BMDL technique performs well, with a lower TRAT of 0.30min and a TEST of 0.10min, respectively. These results provided the superior performance of the ACAD-BMDL method.

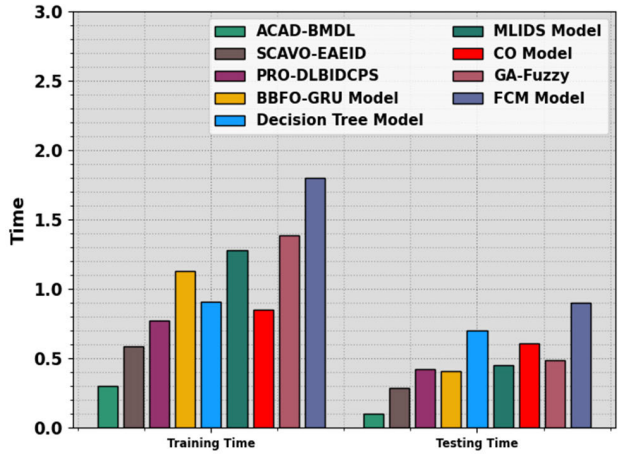

FIGURE 16. TRAT and TEST outcome of the ACAD-BMDL method compared with other models.

TABLE 6. Result analysis of the ablation study under the NSLKDD2015 and CICIDS2017 datasets.

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
NSL-KDD Dataset				
ACAD-BMDL	99.12	99.67	99.30	99.93
BGWO- EESNN	98.59	99.04	98.58	99.41
EESNN Model	98.08	98.37	98.07	98.68
CICIDS2017 Dataset				
ACAD-BMDL	99.36	99.69	99.58	99.80
BGWO- EESNN	98.62	99.1	98.92	99.12
EESNN Model	98.11	98.53	98.23	98.34

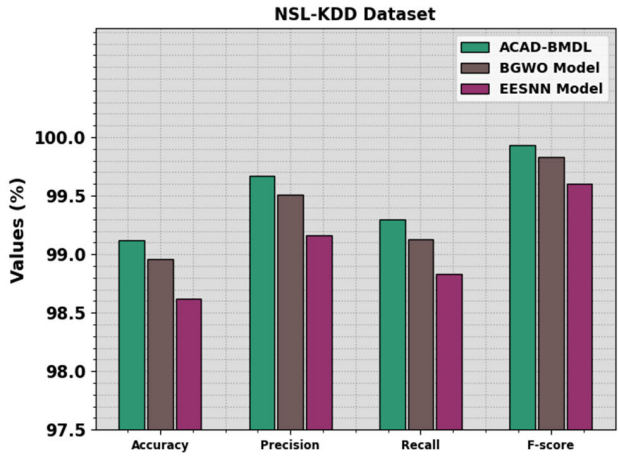

FIGURE 17. Result analysis of the ablation study under the NSLKDD2015 dataset.

Table 6 and Figs. 17-18 illustrates the ablation study of the ACAD-BMDL method under NSLKDD2015 and CICIDS2017 datasets. The ACAD-BMDL model achieved the highest $accu_y$ of 99.12, with $prec_n$, $reca_l$, and F_{score} values of 99.67, 99.30, and 99.93. The BGWO-EESNN model had

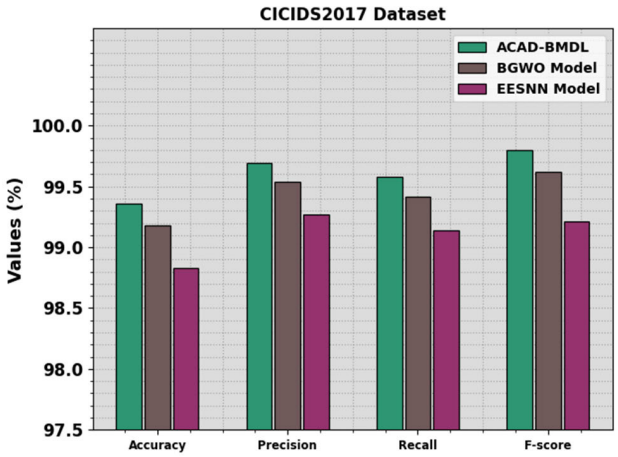

FIGURE 18. Result analysis of the ablation study under the CICIDS2017 dataset.

an $accu_y$ of 98.59, while the EESNN model scored 98.08. On the CICIDS2017 dataset, the ACAD-BMDL model again outperformed others with an $accu_y$ of 99.36, $prec_n$ of 99.69, $reca_l$ of 99.58, and an F_{score} of 99.80. The BGWO-EESNN and EESNN models exhibited accuracies of 98.62 and 98.11, respectively.

V. CONCLUSION

In this study, an ACAD-BMDL method is designed in a CPS framework. The ACAD-BMDL method mainly focuses on improving security in the CPS framework via the cyberattack detection process. The ACAD-BMDL technique involves different sub-processes: Z-score normalization, BGWO-based feature subset selection, EESNN-based classification, and AOA-based hyperparameter tuning. Primarily, the ACAD-BMDL technique underscores Z-score normalization, which is used for scaling the input dataset. In addition, the BGWO technique is applied to select the optimum feature set. Moreover, the EESNN model is exploited for the cyberattack detection. Furthermore, the AOA is exploited to select the optimum hyperparameter for the EESNN method. The empirical analysis of the ACAD-BMDL method is carried out on the benchmark dataset. The performance validation of the ACAD-BMDL method portrays a superior accuracy value of 99.12% and 99.36% under diverse datasets. The limitations of the ACAD-BMDL method comprise potential computational overhead due to the complexity of the BGWO and AOA models. The reliance on Z-score normalization may not address dataset discrepancies or outliers. Future work should optimize computational effectiveness, explore alternative normalization models, and evaluate the technique's performance across several CPS scenarios and datasets. Moreover, integrating real-time adaptive learning mechanisms could enhance the model's robustness against growing cyber threats.

REFERENCES

[1] L. N. H. Pham, "Exploring cyber-physical energy and power system: Concepts, applications, challenges, and simulation approaches," *Energies*, vol. 16, no. 1, p. 42, Dec. 2022, doi: 10.3390/en16010042.

[2] A. O. Alzahrani and M. J. F. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, Apr. 2021, doi: 10.3390/fi13050111.

[3] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini impurity-based weighted random forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, pp. 1–22, Dec. 2022, doi: 10.1186/s42400-021-00103-8.

[4] O. Almomani, "A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system," *Comput., Mater. Continua*, vol. 68, no. 1, pp. 409–429, 2021, doi: 10.32604/cmc.2021.016113.

[5] A. H. Mohammad, T. Alwada'n, O. Almomani, S. Smadi, and N. ElOmari, "Bio-inspired hybrid feature selection model for intrusion detection," *Comput., Mater. Continua*, vol. 73, no. 1, pp. 133–150, 2022, doi: 10.32604/cmc.2022.027475.

[6] M. A. Almaiah, F. Hajjej, A. Ali, M. F. Pasha, and O. Almomani, "A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS," *Sensors*, vol. 22, no. 4, p. 1448, Feb. 2022, doi: 10.3390/s22041448.

[7] A. A. Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," *J. Big Data*, vol. 8, no. 1, pp. 1–19, Dec. 2021, doi: 10.1186/s40537-021-00531-w.

[8] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, Jun. 2020, doi: 10.3390/sym12061046.

[9] X. Huang, J. Liu, Y. Lai, B. Mao, and H. Lyu, "EEFED: Personalized federated learning of execution&evaluation dual network for CPS intrusion detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 41–56, 2023, doi: 10.1109/TIFS.2022.3214723.

[10] R. F. Mansour, "Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment," *Sci. Rep.*, vol. 12, no. 1, p. 12937, Jul. 2022, doi: 10.1038/s41598-022-17043-z.

[11] A. Alqahtani and S. B. Khan, "An optimal hybrid cascade regional convolutional network for cyberattack detection," *Int. J. Netw. Manage.*, vol. 34, no. 5, p. 2247, Sep. 2024, doi: 10.1002/nem.2247.

[12] E. C. Nkoro, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Detecting cyberthreats in metaverse learning platforms using an explainable DNN," *Internet Things*, vol. 25, Apr. 2024, Art. no. 101046, doi: 10.1016/j.iot.2023.101046.

[13] P. B. Dash, H. S. Behera, and M. R. Senapati, "An improved intrusion detection system for the Internet of Medical Things based on deep convolutional neural network," in *Proc. Int. Conf. Comput. Intell. Pattern Recognit.* Singapore: Springer, Jan. 2023, pp. 517–528.

[14] S. H. Chauhdary, M. S. Alkatheiri, M. A. Alqarni, and S. Saleem, "An efficient evolutionary deep learning-based attack prediction in supply chain management systems," *Comput. Electr. Eng.*, vol. 109, Jul. 2023, Art. no. 108768.

[15] P. Joshi, A. Sinha, R. Kundu, R. Shamim, M. K. Bagaria, Y. S. Rajawat, and P. Punia, "AI driven false data injection attack recognition approach for cyber-physical systems in smart cities," *J. Smart Internet Things*, vol. 2023, no. 2, pp. 13–32, Dec. 2023, doi: 10.2478/jsiot-2023-0008.

[16] K. Muthulakshmi, N. Krishnaraj, R. S. Ravi Sankar, A. Balakumar, S. Kanimozhi, and B. Kiruthika, "A novel anomaly detection method in sensor based cyber-physical systems," *Intell. Autom. Soft Comput.*, vol. 34, no. 3, pp. 2083–2096, 2022, doi: 10.32604/iasc.2022.026628.

[17] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni, "TEA-EKHO-IDS: An intrusion detection system for industrial CPS with trustworthy explainable AI and enhanced Krill Herd optimization," *Peer Peer Netw. Appl.*, vol. 16, no. 4, pp. 1993–2021, Aug. 2023.

[18] A. Kamble and V. S. Malemath, "Adam improved rider optimization-based deep recurrent neural network for the intrusion detection in cyber physical systems," *Int. J. Swarm Intell. Res.*, vol. 13, no. 3, pp. 1–22, Jul. 2022.

[19] H. N. AlEisa, F. Alrowais, R. Allafi, N. S. Almalki, R. Faqih, R. Marzouk, M. M. Alnfiai, A. Motwakel, and S. S. Ibrahim, "Transforming transportation: Safe and secure vehicular communication and anomaly detection with intelligent cyber–physical system and deep learning," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1736–1746, Feb. 2024, doi: 10.1109/TCE.2023.3325827.

[20] B. J. Flavia and B. J. Chelliah, "BO-LCNN: Butterfly optimization based lightweight convolutional neural network for remote data integrity auditing and data sanitizing model," *Telecommun. Syst.*, vol. 85, no. 4, pp. 623–647, Feb. 2024.

[21] Y. K. Saheed, A. A. Usman, F. D. Sukat, and M. Abdulrahman, "A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the Internet of Things network," *Frontiers Comput. Sci.*, vol. 5, Apr. 2023, Art. no. 997159.

[22] H. K. Alkahtani, N. Alruwais, A. Alshuhail, N. Nemri, A. B. Miled, and A. Mahmud, "Election-based optimization algorithm with deep learning-enabled false data injection attack detection in cyber-physical systems," *AIMS Math.*, vol. 9, no. 6, pp. 15076–15096, 2024.

[23] B. Antony and S. Revathy, "A novel model for Sybil attack detection in online social network using optimal three-stream double attention network," *J. Supercomput.*, vol. 80, no. 6, pp. 7433–7482, Apr. 2024.

[24] M. Rakhshaninejad, M. Fathian, R. Shirkoohi, F. Barzinpour, and A. H. Gandomi, "Refining breast cancer biomarker discovery and drug targeting through an advanced data-driven approach," *BMC Bioinf.*, vol. 25, no. 1, p. 33, Jan. 2024.

[25] J. Saikam and K. Ch, "EESNN: Hybrid deep learning empowered spatial–temporal features for network intrusion detection system," *IEEE Access*, vol. 12, pp. 15930–15945, 2024, doi: 10.1109/ACCESS.2024.3350197.

[26] M. Nurmuhammed, O. Akdağ, and T. Karadağ, "A novel modified archimedes optimization algorithm for optimal placement of electric vehicle charging stations in distribution networks," *Alexandria Eng. J.*, vol. 84, pp. 81–92, Dec. 2023.

[27] *NSLKDD2015 Dataset*. Accessed: Jun. 6, 2024. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[28] *CICIDS2017 Dataset*. Accessed: Jun. 6, 2024. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[29] L. Almuqren, F. Al-Mutiri, M. Maashi, H. Mohsen, A. M. Hilal, M. I. Alsaid, S. Drar, and S. Abdelbagi, "Sine-cosine-adopted African vultures optimization with ensemble autoencoder-based intrusion detection for cybersecurity in CPS environment," *Sensors*, vol. 23, no. 10, p. 4804, May 2023.

[30] M. Alajmi, H. A. Mengash, H. Alqahtani, S. S. Aljameel, M. A. Hamza, and A. S. Salama, "Automated threat detection using flamingo search algorithm with optimal deep learning on cyber-physical system environment," *IEEE Access*, vol. 11, pp. 127669–127678, 2023, doi: 10.1109/ACCESS.2023.3332213.

ALANOUD AL MAZROA received the M.S. degree in information technology, the M.S. degree in business administration, and the Ph.D. degree in computer science from The Catholic University of America, Washington, DC, USA. She was appointed a member of the following honor societies: Upsilon Pi Epsilon, Delta Epsilon Sigma, and Delta Mu Delta. She is currently an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), Riyadh, Saudi Arabia. She has work experience as a Consultant with The World Bank DC, USA, a Visiting Professor with Marymount University VA, USA, a windows support Engineer with Atos Origin ME and Hewlett-Packard Group Company, and an Operation Specialist with SADAD Payment System, Riyadh. She received the Master's Certificate in Healthcare Informatics from Marymount University, VA, USA.

FAHAD R. ALBOGAMY received the B.Sc. degree (Hons.) in information systems from King Saud University, in 2003, and the M.Sc. and Ph.D. degrees (Hons.) in computer sciences from The University of Manchester, U.K., in 2010 and 2017, respectively. He was the first Dean of the Applied Computer Sciences College, King Saud University. He was a Consultant of academic affairs with the University Vice Presidency for Academic Affairs and Development, Taif University, where he is currently an Associate Professor of computer sciences. He is an Advisor to the President of Saudi Electronic University. His research interests include artificial intelligence, big data, machine learning, NLP, digital image, signal processing, and smart energy.

MOHAMAD KHAIRI ISHAK (Member, IEEE) received the B.Eng. degree in electrical and electronics engineering from International Islamic University Malaysia (IIUM), Malaysia, the M.Sc. degree in embedded system from the University of Essex, U.K., and the Ph.D. degree from the University of Bristol, U.K. He is a registered Graduate Engineer with the Board of Engineers Malaysia (BEM). He is currently an Associate Professor Lecturer in computer engineering with the Department of Electrical and Computer Engineering, Ajman University, United Arab Emirates. His background includes outstanding teaching experience at Universiti Sains Malaysia and Ajman University, instructing students to stimulate engineering information interest and retention while invigorating classes using new technologies and models. His research interests include embedded systems, artificial intelligence, real-time control communications, and the Internet of Things (IoT). He is directed towards developing theoretical and practical methods that can be practically validated. Recently, significant research has been directed towards important industrial issues of embedded networked control systems with AI and the IoT.

SAMIH M. MOSTAFA received the bachelor's and M.Sc. degrees in computer science from the Computer Science-Mathematics Department, Faculty of Science, South Valley University, in 2004 and 2010, respectively, and the Ph.D. degree in computer science from the Advanced Information Technology Department, Graduate School of Information Technology, Kyushu University, Japan, in 2017. He is currently a fellow with the Academy of Scientific Research and Technology (ASRT), Egypt. His research interests include machine learning, the IoT, and CPU scheduling.