IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 October 2024, accepted 15 October 2024, date of publication 17 October 2024, date of current version 5 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3482876

RESEARCH ARTICLE

Modeling of Bayesian-Based Optimized Transfer Learning Model for Cyber-Attack Detection in Internet of Things Assisted Resource Constrained Systems

HAYAM ALAMRO[1], WAHIDA MANSOURI[2], KAWTHER SAEEDI[3], MENWA ALSHAMMERI[4], JAWHARA ALJABRI[5], FAIZ ABDULLAH ALOTAIBI[6], NOHA NEGM[7], AND MAHIR MOHAMMED SHARIF[8]

[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2]Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Northern Border University, Turaif, Arar 91431, Saudi Arabia
[3]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh 25732, Saudi Arabia
[4]Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Saudi Arabia
[5]Department of Computer Science, University College in Umluj, University of Tabuk, Tabuk 71491, Saudi Arabia
[6]Department of Information Science, College of Humanities and Social Sciences, King Saud University, Riyadh 11437, Saudi Arabia
[7]Department of Computer Science, Applied College at Mahayil, King Khalid University, Abha 61421, Saudi Arabia
[8]Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam Bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia

Corresponding author: Mahir Mohammed Sharif (m.adam@psau.edu.sa)

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/86/45. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R361), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2024R838), King Saud University, Riyadh, Saudi Arabia. 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-2024-2899-01. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

ABSTRACT Security donates itself as one of the largest attacks on the support and development of the Internet of Things (IoT). Security challenges are evident in cyber-security threads that direct the main Internet service provider and weaken a significant part of the complete Internet by benefiting from defective and vulnerable embedded gadgets. Numerous devices inhabit at-home systems with user-administrators unfamiliar with network security best practices, creating simple goals for the attackers. So, security solutions are required to direct the untrusted and insecure public networks by mechanizing defences over affordable and nearby direct network data sharing. The growth of automatic cyberattack classification and detection tools utilizing artificial intelligence (AI) and machine learning (ML) devices become vital to achieving safety in the IoT environment. It is desired that safety issues allied to IoT devices be effectively diminished. This article proposes an Advanced Ensemble Transfer Learning for Cyberthreat Detection in Low Power Systems (AETL-CDLPS) technique. The primary intention of the AETL-CDLPS technique is to automate the detection of cyber-attacks for IoT-assisted resource-constrained systems. The AETL-CDLPS technique utilizes a linear scaling normalization (LSN) model to normalize the input data. Next, the AETL-CDLPS technique employs an improved coati optimization algorithm (ICOA)-based feature selection technique to choose optimal features. For the cyber threat detection process, an ensemble transfer learning (TL) model comprises three classifiers, namely gated recurrent Unit (GRU), deep convolutional neural network (DCNN), and stacked sparse autoencoder (SSAE). Finally, the Bayesian optimization algorithm (BOA) is utilized to optimize the hyperparameter tuning of the three ensemble techniques. The AETL-CDLPS model's performance validation is performed using the Bot-IoT dataset. The comparison study of the AETL-CDLPS

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

method portrayed superior Accuracy, Precision, Recall, and F-Score values of 99.19%, 96.10%, 95.97%, and 96.03% over existing models.

INDEX TERMS Ensemble transfer learning, cyberthreat detection, low power systems, improved coati optimization algorithm, Internet of Things.

I. INTRODUCTION

The Internet of Things (IoT) is a developing industry that can drastically transform the utilization of the real environment and technology [1]. There should be 30 billion IoT-associated devices in 2025. With the rise of associated gadgets, data breaches also happen because of IoT devices' frequently short processing and storage abilities. Subsequently, IoT devices often have restricted calculation storage and power; they could be vulnerable to attackers as IoT develops [2]. Developments in mobile technology have preceded the IoT growth, restructuring areas containing smart cities, healthcare, and real estate. These connected devices are smart gadgets that utilize lightweight central processing units and network interface cards achieved by various interface services. The IoT could influence our combined future as technology grows substantially [3]. The fast development of IoT has created security for major crucial problems in a network-dependent system. Viruses, Hackers, and other damaging software can weaken the safety and data consistency. Furthermore, data insecurity can instantly undermine the overall IoT's security and initiate various dangerous conditions. Robust IoT security solutions are in great demand. With the rise of IoT device development and novel threads, it would be progressively significant to analyze and gather data to preserve the security of these devices [4]. With the growth in the count of IoT devices, a rising frequency of cyber risks exists, raising significant security challenges for various networks and methods. As adversaries constantly change their strategies, the requirement for innovative and effectual recognition methods develops supreme [5].

Malicious or malware software can be set on a computer to disturb its function and damage electrical data. Viruses, Trojan horses, ransomware, spyware, worms, adware, and malvertising are significant malware forms. Malign intrusions on computer gadgets and networks are other cyber-attacks in cyberspace. This intrusion is used for scanning and identifying the vulnerabilities of a computer system or network [6]. An intrusion detection system (IDS) was utilized to defend against nearby intrusion. Machine learning (ML) has recently created significant development as device intelligence has advanced from a lab curiosity to realistic machinery with numerous critical applications. IoT device intellect offers essential solutions to novel or zero-day threads; such devices could be controlled [7]. To defend data traversing in IoT systems, service suppliers approve mitigation and identify methods to recognize potential security breaches. With the development of Artificial Intelligence (AI) and ML, this method could forecast an arriving cyberattack. Utilizing robust data exploration techniques (ML), the ''abnormal'' and ''normal'' behaviours of IoT gadgets and modules in their atmosphere are identified [8]. These techniques are crucial for changing the IoT system's security in a security-based smart network and not just securing device connections. The rapid expansion of connected devices presents essential data security and privacy threats. As IoT systems proliferate, they become attractive targets for cyber-attacks due to their limited processing power and storage capabilities [9]. This vulnerability is specifically concerning in critical sectors, namely healthcare and smart infrastructure, where compromised devices can result in severe consequences. Developing robust detection mechanisms that can operate efficiently within these constraints is crucial. Therefore, exploring innovative methods, namely, Bayesian optimized TL, can substantially improve the resilience of IoT systems against growing challenges [10].

This article proposes an Advanced Ensemble Transfer Learning for Cyberthreat Detection in Low Power Systems (AETL-CDLPS) technique. The primary intention of the AETL-CDLPS technique is to automate the detection of cyber-attacks for IoT-assisted resource-constrained systems. The AETL-CDLPS technique utilizes a linear scaling normalization (LSN) model to normalize the input data. Next, the AETL-CDLPS technique employs an improved coati optimization algorithm (ICOA)-based feature selection technique to choose optimal features. For the cyber threat detection process, an ensemble transfer learning model comprises three classifiers, namely gated recurrent Unit (GRU), deep convolutional neural network (DCNN), and stacked sparse autoencoder (SSAE). Finally, the Bayesian optimization algorithm (BOA) is utilized to optimize the hyperparameter tuning of the three ensemble techniques. The AETL-CDLPS model's performance validation is performed using the Bot-IoT dataset. The key contribution of the AETL-CDLPS model is listed below.

- The AETL-CDLPS approach utilizes the LSN model to standardize input data, improving the accuracy and convergence of subsequent models. This methodology confirms that the overall features are treated equally, mitigating bias and enhancing model performance. Using LSN, the framework efficiently prepares the data for more robust evaluation and detection.
- The AETL-CDLPS approach utilizes the ICOA technique for effective feature selection, which crucially mitigates dimensionality while conserving substantial data. This methodology reorganizes the dataset, improving the model's capability to focus on the most relevant features. As a result, the overall performance and accuracy of the cyber threat detection process are enhanced.

- An ensemble TL model incorporates GRU, DCNN, and SSAE to improve cyberthreat detection across various scenarios. This integration utilizes the merits of every method, enhancing detection accuracy and robustness. By addressing diverse attack types and conditions, the ensemble technique confirms the system's overall safety.
- The BOA technique employs the AETL-CDLPS method for hyperparameter tuning, optimizing the performance of the ensemble classifiers. This methodology improves the model's efficiency by systematically exploring hyperparameter combinations to detect the best settings. As a result, the classifiers attain enhanced accuracy and efficiency in cyberthreat detection.
- The AETL-CDLPS approach presents an integrated model combining normalization, feature selection, ensemble learning, and hyperparameter optimization. By restructuring these processes, it efficiently tackles the complexities of cyber threat detection within a cohesive framework. The novelty is in its holistic design, which improves the detection system's robustness and adaptability across diverse scenarios, enhancing overall accuracy and response times.

II. LITERATURE OF WORKS

Salim et al. [11] propose a new Cyber threat Detection Method for IoT systems, which utilizes Digital Twin technology and an enhanced FL method. Our theory suggests incorporating Digital Twin methods in an IoT security structure to strengthen actual cyber threat detection abilities. The author executes an 'Adaptive Thresholding with Initial Preventing technique' based approach in FL to consistently aggregate and train local methods depending on pre-defined training circles, thus assuring that all local methods promote the global method till a target precision is attained. AboulEla et al. [12] study cybersecurity in the IoMT context, which contains cybersecurity methods utilized for many healthcare devices associated with the method. This research aims to summarise various AI-based techniques and methodologies and investigate the related solution methods used in cybersecurity for medicare methods. The studied methods are additionally classified into four categories: DL and ML methods, integration of DL and ML methods, Transformer-based, and other advanced methods containing graph-based techniques and blockchain techniques. Bergies et al. [13] introduce a new IoT structural model incorporating model predictive control (MPC) and DNN structures. The presented structure goals are to improve AEV performance and empower them to counteract the disruptive impact of erroneous data intrusions resulting from cyber breaches. This research enhances the structure by including trajectory prediction to ensure the early detection of possible threats without impacting privacy thoughts. Rookard and Khojandi [14] present RRIoT to alleviate this problem. RRIoT uses a Deep Deterministic Policy Gradient RL system in combination with a layer of LSTM in an adversarial atmosphere to identify and detect threads. The author assesses our technique versus new and advanced RL / ML methods created upon preceding RL methods like DDPG, DQN, and DDQN. Our outcomes specify that our presented RRIoT usually executes more than current ML methods and performs greater than new RL methods by novel network structures.

Rana et al. [15] present Cybersecurity threat identification in IoT by utilizing KDNN-SAE. In the presented method, the data preprocessing measure was initially performed in the essential growth, separating the dataset into two sections: testing and preparing. At that time, flow-based features have been removed from the preprocessed data. By that time, the features to be used by the systems are selected in the feature identification using the GA. Finally, our method ends with the implementation of the ML method KDNN-SAE. Aldhaheri et al. [16] examine cutting-edge intrusion detection techniques for IoT security based on DL. The author studies current innovations in IDS for IoT, emphasizing the fundamental DL methods, evaluation metrics, associated datasets, and types of attacks. Moreover, the author debates the challenges of using DL for IoT security and recommends possible fields for upcoming studies. In [17], an ensemble DL technique, which utilizes the benefits of the AE and the LSTM structure to detect out-of-norm actions for cyber threat searching in IIoT is presented. In this method, the LSTM generates a method on the usual time data sequence (present and past data) to study usual data patterns. AE recognizes the significant data features to decrease data size. Additionally, the unbalanced nature of IIoT databases could not studied in the utmost of the preceding article, impacting lower performance and precision. Alrowais et al. [18] present a novel MFO with RELM method, called MFO-RELM, for Cyber-attack risk classification and Detection from the IoT atmosphere. The suggested MFORELM method effectively recognizes cyber security threats in the IoT atmosphere. The MFO-RELM method preprocesses the real IoT data into an expressive format to achieve this. Additionally, the RELM method obtains the preprocessed data and performs the classifier method. Sana et al. [19] present a novel IDS. Also, supervised ML and DL models, comprising LSTM and vision transformers, utilize Bayesian optimization to enhance performance. Wang et al. [20] propose ResADM, a transfer-learning (TL)-based attack detection methodology. It balances attack sample dispersion through intentional sampling, extracts key features utilizing importance-based selection, and employs a ResNet-based network for optimizing source model parameters for detection.

Ioannou et al. [21] classify attacks through anomaly detection with model retraining and utilize Federated Learning (FL) to keep the cloud server updated while conserving data privacy. Shafin et al. [22] propose a lightweight multiclass malware detection approach for embedded devices, integrating CNNs for feature learning with bidirectional LSTM for temporal modelling to identify recent malware. Jaradat et al. [23] employ a Long Short-Term Memory (LSTM) network with conventional ML models, namely Random Forest (RF), Support Vector Machine (SVM), and K-Nearest

Neighbor (kNN), to classify cyberattacks. Willeke et al. [24] develop and experiment with a data-efficient FL framework for IoBT settings for intrusion detection utilizing only raw network traffic in restricted, resource-limited environments. Asgharzadeh et al. [25] propose FECNNIoT, a convolutional neural network (CNN) for anomaly detection in IoT, and develop BMEGTO, an improved Gorilla troops optimizer (GTO) model for efficient feature selection. Integrating FECNNIoT, BMEGTO, and KNN classification methods results in a hybrid method, namely CNN-BMEGTO-KNN. Javeed et al. [26] suggest a deployment architecture for the proposed CUDA-powered IDS using OpenStack Tacker in a real SDN environment. AboulEla et al. [12] explore cybersecurity in the Internet of Medical Things (IoMT), classifying AI-based methods into ML, DL, hybrid models, and advanced methodologies such as graph-based and blockchain. Tiwari and Waoo [27] introduce an IoT-based smart home cyber-attack detection. It employs ML and network traffic analysis to detect real-time attacks and comprises automated defences, isolating compromised devices while allowing homeowners to customize security policies. Zukaib et al. [28] propose a novel meta-IDS (Meta-IDS) that utilizes a meta-learning model. This method incorporates signature-based and anomaly-detection models while integrating privacy-preserving methods for sensitive IoMT data. Verma and Chandra [29] present the RepuTE Framework for the Fog-IoT domain, employing a soft voting ensemble learning method. Furthermore, a novel feature selection model is presented to detect relevant features in advance.

The existing cyber threat detection methodologies employing Digital Twin technology and improved FL may increase system complexity and demand crucial computational resources, limiting deployment in resource-constrained environments. Moreover, while summarizing AI models for IoT, growing methods and practical implementation threats in healthcare may need to be addressed. Integrating model predictive control with deep neural networks (DNNs) can result in high computational overhead, hindering real-time applications. Additionally, lightweight malware detection models might sacrifice accuracy for efficiency, risking the omission of advanced threats. Other frameworks may face data imbalance, encounter integration threats, and require extensive validation to confirm efficiency across diverse IoT scenarios, specifically in real-time threat detection and response. Despite enhancements in cybersecurity for IoT and related systems, there still needs to be a gap in addressing detection models' scalability and computational effectualness in resource-constrained environments. Furthermore, many existing frameworks need to pay more attention to incorporating growing AI methods and their practical implementation threats in real-world scenarios, specifically in healthcare and smart home settings.

III. MATERIALS AND METHODS

In this article, we present a design of the AETL-CDLPS technique. The major aim of the technique lies in the automated detection of cyber-attacks for IoT-assisted resource-constrained systems. To achieve this, the AETL-CDLPS technique contains various stages, such as data preprocessing, feature selection, data transfer using ensemble transfer learning, and hyperparameter tuning. Fig. 1 represents the workflow of the AETL-CDLPS technique.

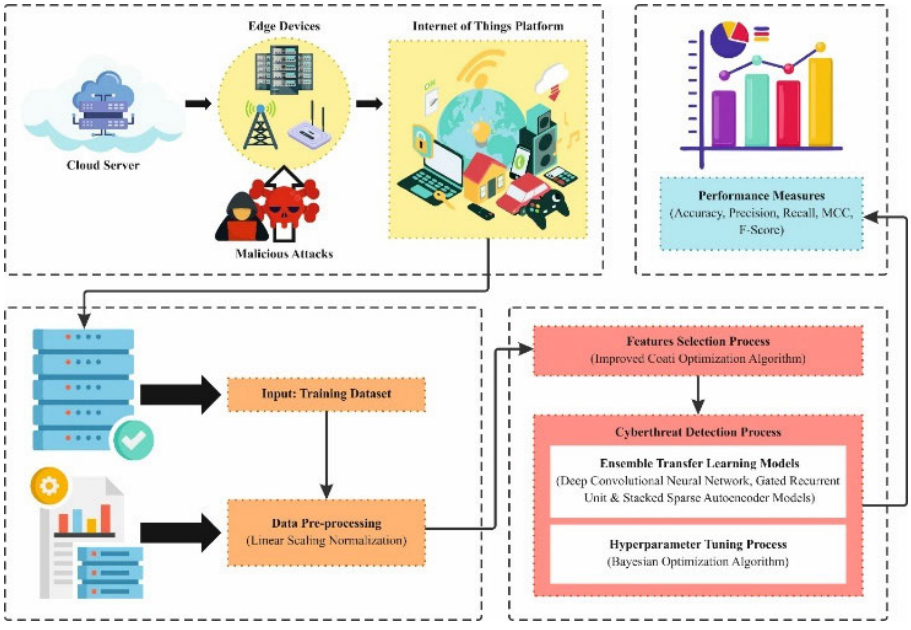

FIGURE 1. Workflow of AETL-CDLPS technique.

A. LINEAR SCALING NORMALIZATION

Initially, the AETL-CDLPS technique utilizes the LSN model to normalize the input data [30]. The LSN model is a data pre-processing approach that regularly scales features to a particular range [0, 1] to guarantee constant input for ML methods. This model is chosen for its simplicity and efficiency in adjusting the input data to a standard scale without distorting the relationships between the values. This methodology improves the convergence speed of ML models, making it specifically appropriate for models sensitive to the scale of input features, namely neural networks. The LSN method also confirms that all features contribute equally to the model training, mitigating bias and enhancing overall performance. Its straightforward implementation also allows for easy interpretation and maintenance of the normalization process compared to more complex techniques.

During the detection of cyber threats, LSN aids in normalizing network traffic data, which might take broadly changing scales, to increase the performance and accuracy of detection algorithms. By bringing all featured values to a uniform scale, LSN decreases the impact of anomalies and guarantees that no solitary feature excessively affects the prediction models. These can be especially significant in cyber threat detection methods, whereas several features like protocol types, packet size, and time intervals may differ significantly. LSN can be regularly used before serving data into DL or ensemble technique to improve their capacity to identify subtle patterns indicative of threads.

B. ICOA-BASED FEATURE SELECTION

Next, the AETL-CDLPS technique uses the ICOA-based FS to elect optimal features [31]. Fig. 2 illustrates the workflow of the ICOA approach. The model also presents various merits for feature selection, primarily through its capability to balance exploration and exploitation in the search space, resulting in more efficient detection of relevant features. Its adaptive mechanism improves convergence speed and accuracy compared to conventional approaches, mitigating dimensionality while maintaining classification performance. ICOA also reduces problems such as overfitting by choosing only the most informative features, making it particularly suitable for high-dimensional datasets. This results in improved model interpretability and reduced computational costs, positioning ICOA as a superior choice for feature selection tasks. In the optimization method, the first population selection significantly affects the last optimization outcome. Conventional COA generally implements a random initialization technique to create the first population. However, this technique can cause inadequate population diversity, impacting the method's global search ability. To overwhelm this problem, we present a novel initialization technique. Especially for every distinct in the population, we initially create a randomly generated value $x(i)$ among 0 and 1. Next, we utilize a Sine map to transform this randomly generated number into a chaotic sequence. This method is signified by Eq. (1).

$$F_i = \sin(\pi x(i)) \tag{1}$$

On the other hand, F_i denotes individual x Sine map value.

Subsequently, they utilize the created chaotic series to make the first individuals, which satisfy the specific conditions. For the size of every individual, it utilizes Eq. (2) to create a value among the value range b_j^{lower} and b_j^{upper}.

$$x_{i,j} = b_j^{lower} + F_i\left(b_j^{upper} - b_j^{lower}\right) \tag{2}$$

whereas $x_{i,j}$ signifies the dimension j digit after mapping the individual i, b_j^{upper} signifies dimension j upper limit, and b^{ower} represents the dimension j lower limit.

Eventually, it uses the random and Gaussian Walk tactic to enhance the first individuals. Every size of an individual can be created with a number at random p. If $p \geq 0.5$; it can be selected to utilize the Gaussian Walk tactic for the optimizer, as displayed in Eqs. (3) and (4).

$$\Delta = N(0, 1) \times \min\left(\left(b_j^{upper} - x_{i,j}\right), \left(x_{i,j} - b_j^{lower}\right)\right) \tag{3}$$

$$x_{i,j} = x_{i,j} + \Delta \tag{4}$$

$N(0, 1)$ denotes a randomly generated number after the environmental distribution between 1 and 0.

When $p < 0.5$, they select the random walk tactic for the optimizer, as displayed in Eqs. (5) and (6).

$$\Delta = U(-1, 1) \times \min\left(\left(b_j^{upper} - x_{i,j}\right), \left(x_{i,j} - b_j^{lower}\right)\right) \tag{5}$$

$$x_{i,j} = x_{i,j} + \Delta \tag{6}$$

whereas $U(1, -1)$ signifies a uniform distribution among [1, -1].

We combine the individuals enhanced through the random walk strategies, Gaussian Walk, and chaotic mapping and choose the 1st N individuals as the first population. Through this technique, they can create a first population through a great range, thus enhancing the system's global searching abilities.

1) SUBPOPULATION DIVISION

After creating the first population, we divide it into dual sub-populations. This partition tactic aims to intensify the population diversity, improving the ICOA global search ability. Every subpopulation will independently implement the following optimization method. We first classify based on the fitness of every individual. Consequently, we combine the lowest and highest fitness and allot this couple of individuals in the 1st sub-population. Later, we combine the 2nd-greatest and 2nd-poor performing individuals and allot this couple of individuals in the 2nd sub-population. We iterate this method till every individual has been allocated to dual subpopulations.

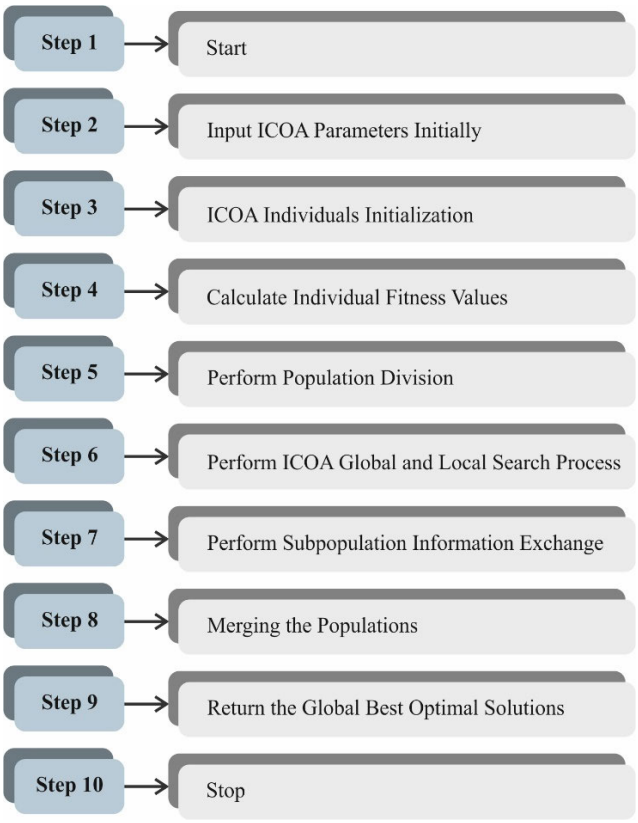

FIGURE 2. Workflow of ICOA technique.

2) GLOBAL SEARCH PHASE

After separating the first population into double subpopulations, we initiate the upgrade coati population process within the search space. The initial stage of this method is to imitate the tactic while coatis attack iguanas, which is the stage of global searching. Once the iguana drops in the field, the coatis attack and kill it. This tactic alerts the coatis to move to another location within the searching space, so representing the COA optimizer method retains explorative abilities in the problem space global search. During the COA method, the location of the finest population participant represents the iguana location. There is a perception that 50 % of coatis would ascend the tree, but the other 50% should expect the iguana to slope down. Then, we could pretend the location over the following numerical method:

$$X_{i,j}^{P1} = x_{i,j} + r\left(G_j - Ixi, j\right), i = 1, 2, \cdots, \left\lceil \frac{N}{2} \right\rceil \quad (7)$$

whereas $X_{i,j}^{P1}$ represents the novel location of coati i in dimension j, r denotes a randomly generated number among 0 and 1, G_j signifies the iguana location in the dimension j, I represents a random number chosen from the set 1, 2}, N denotes coati counts, and $\left\lceil \frac{N}{2} \right\rceil$ represents a leading number of not more than $\frac{N}{2}$.

If the iguana cascades, it is located arbitrarily within the searching space. Given these randomly formed locations, the coats by the ground will transfer during the search space. Eqs. (8) and (9) equate this phase.

$$G_j^g = b_j^{lower} + r(b_j^{upper} - b_j^{lower}) \quad (8)$$

Meanwhile, G_j^g denotes the iguana's location in dimension j on the ground.

$$x_{i,j}^{P1} = \begin{cases} x_{i,j} + r\left(G_j^g - Ix_{i,j}\right), & if\ F_{G^g} \le F_j \\ x_{i,j} + r\left(x_{i,j} - G_j^g\right), & otherwise \end{cases}$$
$$i = \left\lceil \frac{N}{2} \right\rceil + 1, \left\lceil \frac{N}{2} \right\rceil + 2, \cdots, N \quad (9)$$

whereas F_{G^g} represents the iguana's objective function value afterwards down in the ground, F_j signifies the value of an objective function of coati i.

To monitor the hunt method finer, evade dropping into local targets, enhance searching efficacy, and enhance the solution qualities, they enhanced the conventional COA method and proposed a weight factor w depending on the adaptive factor. The w computation is presented in Eq. (10).

$$w_{i,j} = w_{\min} + f_{i,j} \cdot (w_{\max} - w_{\min}) \quad (10)$$

whereas $w_{i,j}$ represents *ith* coati weight factor in size j, and $f_{i,j}$ denotes *ith* coati adaptive factor in size j, which is computed by Eq. (11).

$$f_{i,j} = \alpha \cdot \frac{X_{best,j} - X_{current,i,j}}{X_{best,j} - X_{worst,j}} + \beta \cdot \frac{X_{best,j} - X_{current,i,j}}{X_{best,j}} \quad (11)$$

whereas $f_{i,j}$ denotes an adaptive factor of *ith* coati in *jth* dimension. $X_{best,j}$ signifies the finest solution location in dimension j. $X_{current,i,j}$ represents the existing location of the *ith* coati in *the jth* dimension. $X_{worst,j}$ means poor solution value location in *jth* size. α and β are constants utilized to fine-tune the adaptive factor impact.

Next, the ICOA search at this phase is displayed as Eqs (12) and (13).

$$X_{i,j}^{P1} = x_{i,j} + w_{i,j} \cdot r\left(G_j - Ixi, j\right), i = 1, 2, \cdots, \left\lceil \frac{N}{2} \right\rceil,$$
$$j = 1, 2, \cdots, m \quad (12)$$

$$x_{i,j}^{P1} = \begin{cases} x_{i,j} + w_{i,j} r\left(G_j^g - Ix_{i,j}\right), & if\ F_{G^g} \le F_i \\ x_{i,j} + w_{i,j} r\left(x_{i,j} - G_j^g\right), & otherwise \end{cases}$$
$$i = \left\lceil \frac{N}{2} \right\rceil + 1, \left\lceil \frac{N}{2} \right\rceil + 2, \cdots, N \quad (13)$$

Through this, we could effectively imitate the tactic of coatis attacking iguanas, therefore attaining an effective searching space study within the global searching phase. These techniques not only enhance population diversity but also global search ability.

Next, we equate the upgraded individuals to the original individuals. When the upgraded individuals have been enhanced and upgraded, the existing body, or else, retains the position. The mathematical formulation is represented below:

$$X_i = \begin{cases} X_i^{P1}, & if\ F_i^{P1} \le F_i \\ X_i, & otherwise \end{cases} \quad (14)$$

whereas F_i^{P1} represents *the ith* coati objective function value at the novel location, F_j denotes the *ith* coati objective function value at the earlier location.

3) LOCAL SEARCH

The 2nd stage of upgrading the location of coatis within the searching space represents the avoiding predator method and is also denoted the exploitation phase. This phase is embedded in the coati's natural behaviour while they meet predators and run away from theirs. If a predator attacks the coati, it should be suspended from its existing location. This tactic instructs the coati to a safer position near its existing location, representing the COA process's ability in local hunting.

To pretend these behaviours, create an arbitrary location close to the location of every coati. Especially the local upper-bound $b_{j,U}^{loc}$, and local lower-bound $b_{j,L}^{loc}$ of every decision variable is represented in Eq. (15).

$$b_{j,lower}^{loc} = \frac{b_j^{lower}}{t}, b_{j,upper}^{loc} = \frac{b_j^{upper}}{t}, t = 1, 2, \cdots, T \quad (15)$$

whereas $b_{j,lower}^{loc}$ *and* $b_{j,upper}^{loc}$ denotes local lower and upper limits of the jth decision variable, t signifies several iterations, and T denotes the maximum iteration numbers.

After that, every individual is upgraded based on Eq. (16).

$$X_{i,j}^{P2} = x_{i,j} + (1 - 2r)\left(b_{j,lower}^{loc} + r\left(b_{j,upper}^{loc} - b_{j,lower}^{loc}\right)\right)$$
$$i = 1, 2, \cdots N, \quad (16)$$

whereas $X_{i,j}^{P2}$ signifies the novel location of coati i in dimension j.

When the upgraded individual is superior, the existing individual is upgraded or the status quo is retained, as displayed in Eq. (17).

$$X_i = \begin{cases} X_i^{P2}, & if\ F_i^{P2} \leq F_i \\ X_i, & otherwise \end{cases} \quad (17)$$

whereas F_i^{P2} represents a value of an objective function of *ith* coati on the novel location, F_j denotes a value of a main function of *ith* coati on the preceding location, and X_i represents the original location of coati i.

In ICOA methodology, the objectives can be joined as a single objective equation such that a present weight recognizes every objective position. A fitness function (FF) that integrates both FS objectives can be implemented during this work, as described in Eq. (18).

$$Fitness\,(X) = \alpha \cdot E\,(X) + \beta * \left(1 - \frac{|R|}{|N|}\right) \quad (18)$$

In which $Fitness(X)$ defines the fitness rate of subset X, $E(X)$ implies the classifier error values by employing the particular features within the X subset, $|R|$ and $|N|$ indicate the elected feature counts and the new feature counts within the database correspondingly, α and β represents the weights of the classifier error and reduction ratio, $\alpha \in [0, 1]$ and $\beta = (1-\alpha)$.

C. ENSEMBLE TRANSFER LEARNING

Next, for the cyber threat detection process, an ensemble transfer learning model involves three classifiers, DCNN, GRU, and SSAE, chosen for their complementary merits in cyber threat detection. The DCNN model outperforms in capturing spatial features in data, making it efficient for image-like inputs. At the same time, GRU is adept at modelling temporal dependencies, which is significant for analyzing sequential attack patterns. The SSAE method improves feature extraction and dimensionality reduction, facilitating a better representation of intrinsic data. This incorporation allows for enhanced accuracy and robustness in threat detection, implementing the unique capabilities of each model to address diverse attack vectors effectually.

1) DEEP CONVOLUTIONAL NEURAL NETWORK

The DCNN structure begins its early convolutional layer with a definite 7 × 7 convolution (Conv) kernel, which is tracked by a following max−pooling layer [32]. By monitoring this, dual Conv layers were used, both mixed and single 5 × 5 and 3 × 3 Conv kernels, achieved by one more max−pooling layer, resulting in a configuration including 3 max−pooling layers and six stacked Conv layers. The diverse Conv kernel helps in the extraction of features through numerous dimensions and decreases connection parameters among neurons. Investigating every feature map from the 2nd max−pooling and 3rd Conv layers establishes the impact of Conv kernel dimension and stride on the output size of the feature map. During this work, 2 strides were executed for the early Conv layer and 1st max-pooling layer, while following pooling and Conv layers uphold a stride of 1. To safeguard stability in resultant feature mapping for size 2 × 2 and Conv layers with 5 × 5 kernel padding is combined. Moreover, pooling processes are employed to determine the effect of feature map resolution and accurate position, thus alleviating over-fitting and upholding the network's detection efficiency. Fig. 3 represents the DCNN structure

DCNNs organize recognition concepts into dual foremost layers such as pooling (sub-sampling) and Conv. The k^{th} feature map f_m is signified as f_{mij}^k utilizing tanh with connection weights $w^k\lambda$ and biases $b^k\lambda$, as definite in Eq. (19).

$$f_{mij}^k = \tanh\left(\left(w^k \cdot x\right)_{ij} + w^k\right) \quad (19)$$

The subsampling layer generates spatial invariance by declining f_m,s resolution. Each f_m looks like one in the preceding layer. Eq. (20) summarizes the model employed to define the pooling procedure.

$$\lambda_j = taM\left(W \sum_{M \cdot M} \lambda_i^{m \cdot m} + \omega\right) \quad (20)$$

whereas $\lambda_i^{m \cdot m}$ signifies an input, ω and W denotes bias and trainable scalar, correspondingly. These parameters were required to be altered to improve the rate of accuracy.

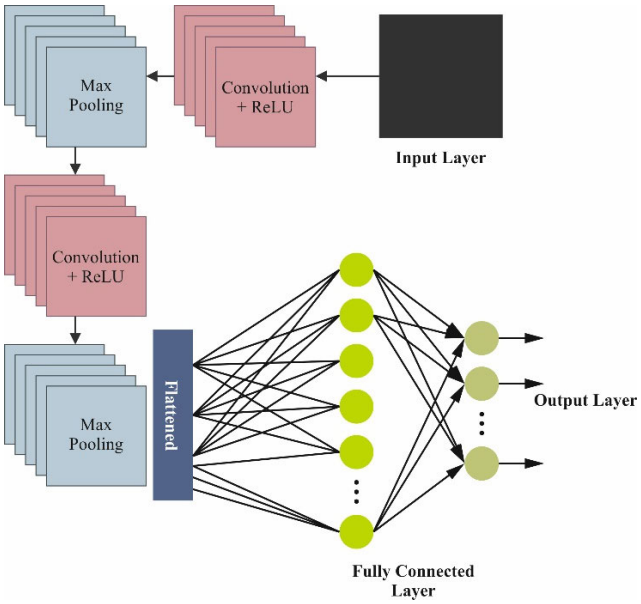

FIGURE 3. Structure of DCNN.

2) GATED RECURRENT UNIT

In contrast with the RNN network structure, GRU presents dual gate units for the particular keeping and forgetting of input information, such as update and reset gates [33]. Fig. 4 demonstrates the architecture of the GRU model.

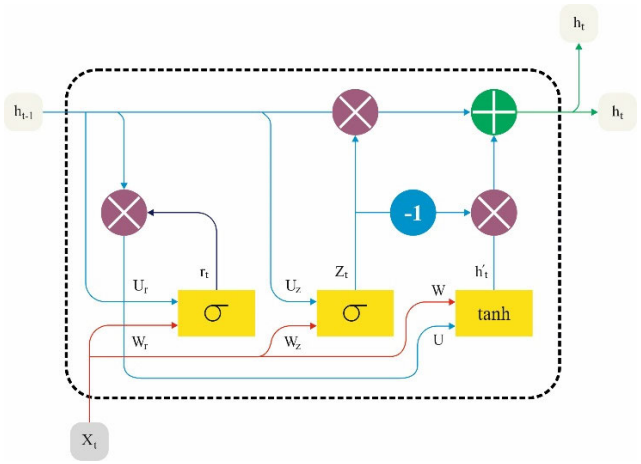

FIGURE 4. The architecture of the GRU method.

The formulation of the realization process is mentioned in Eq. (21).

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{21}$$

Here, σ denotes the sigmoid function, h_{t-1} means cell state at the preceding time, x_t refers to an input value at present, and W_z and W_r specify weight parameters of the update and reset gates, correspondingly.

After attaining the reset factor r_t, the signal of the reset gate is multiplied with h_{t-1} and united with an input. The data are transformed into $(-1, 1)$ utilizing the activation function of tanh to get the selected cell h_t candidate layer. At last, input data is upgraded by an update factor z_t of the update gate. Its mathematical formulation is mentioned below in Eq. (22).

$$\tilde{h}_t = \tanh\left(W_h \cdot [r_t * h_{t-1}, x]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{22}$$

whereas W_h means the weight parameter.

This mostly comprises three phases: learning the existing state, reset input data, and selecting and forgetting the memory. When equated through LSTM, the GRU effect is not dissimilar, but its computation is very simple, and the model's training is highly effective. The fault recognition method is dependent upon GRU prediction error, and the Pauta condition is exposed in Eq. (23).

$$\hat{Y} = W_f h_t + b_f \Rightarrow \begin{cases} If \left|\hat{Y} - Y\right| \leq 3\sigma \Rightarrow Normal \\ If \left|\hat{Y} - Y\right| > 3\sigma \Rightarrow Abnormal \end{cases} \tag{23}$$

Here, w_f and b_f denote fully connected (FC) layer parameters. $\hat{Y}$ represents the forecast's value.

3) STACKED SSAE

AE are artificial neural networks (ANN) trained utilizing an unsupervised method that initially targets studying coded data representations, compressing the input into a latent space representation, and then recreating the data over this compressed data [34]. The primary structure of this kind of network contains three parts: the Decoder, the Encoder, and the Latent Space. Fig. 5 shows the structure of the SSAE methodology.

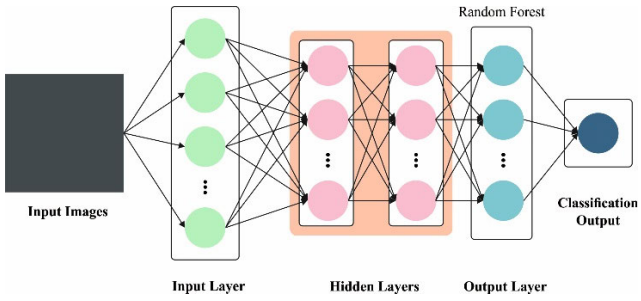

FIGURE 5. Structure of the SSAE technique.

The encoding and decoding methods (Eq. (24)) and (Eq. (25)) are represented below:

$$H = \sigma_1\left(W_{ij} \cdot X + b_1\right) \tag{24}$$
$$\hat{X} = \sigma_2\left(W_{jk} \cdot H + b_2\right) \tag{25}$$

whereas W_{ij}, W_{jk} are the weights connection matrices among the hidden layer (HL)-output layer and input layer-HL, $X = (X_1, X_2, \ldots, X_n)$ represents the data of the input vector, $H=(H_1, \ldots, H_m)$ denotes low dimensional vector and $\hat{X}=(\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n)$ signifies output data vector. b_1, b_2 are

bias vectors and σ_1, σ_2 are the activation functions utilized in every part?

The weight matrix and bias vectors optimization is implemented by minimizing a function of error, which states the variance among the input X and the input $\hat{X}$ reconstruction, as displayed in Eq. (26). This function utilizes the mean squared error function, whereas N denotes the number of input samples.

$$E(W, \sigma) = \frac{1}{N} \sum_{j=1}^{N} ||\hat{X} - X||^2 \tag{26}$$

In the SAE, to generate the much more effective coding, dual regularization terms, regularisation, and weight sparse penalty terms, are included in the error function, affecting only a neuron sub-set to be stimulated. The average neuron activation in the HL $\hat{\rho}$ is specified by:

$$\hat{\rho} = \frac{1}{N} \sum_{j=1}^{N} z_i(x(j)) = \frac{1}{N} \sum_{j=1}^{N} \sigma(w_{i^T} \cdot x_j + b_i) \tag{27}$$

As the average activation $\hat{\rho}$ is meant to be near to constant ρ and generally a tiny positive number near 0, the Kullback-Leibler (KL) divergence is utilized.

$$\Omega_{sparse} = KL(\rho||\hat{\rho}) = \rho \cdot \log\left(\frac{\rho}{\hat{\rho}_i}\right) + (1-\rho) \cdot \log\left(\frac{1-\rho}{1-\hat{\rho}_i}\right) \tag{28}$$

In the same way, a term of weight regularisation, named L2 regularisation, is also added to evade network overfitting.

$$\Omega_{weights} = \frac{1}{2} \sum_{l=1}^{L} \sum_{j=1}^{n_l} \sum_{i=1}^{k_l} (w_{ji}^l)^2 \tag{29}$$

whereas L represents the number of HLs, n_l is the layer one output size, *and* k_l is the layer 1 input size.

So, the SAE cost function is specified below:

$$E_{sparse}(W, \sigma) = E(W, \sigma) + \mu \cdot \Omega_{sparse} + \beta \cdot \Omega_{weights} \tag{30}$$

whereas μ signifies the co-efficient for the L2 term of regularisation and β the co-efficient for the sparsity term of regularisation.

D. HYPERPARAMETER TUNING METHOD

Finally, the BOA method is used for the optimal hyperparameter tuning of the three ensemble techniques [35]. This model was chosen for its effectiveness in exploring hyperparameter space. Unlike conventional grid or random search methods, BOA utilizes a probabilistic model to select the next set of hyperparameters intelligently, balancing exploration and exploitation. This results in faster convergence to optimal values, improving the performance of the model without extensive computational costs. Furthermore, BOA is specifically efficient in scenarios where evaluations are costly, making it ideal for tuning complex ensemble models

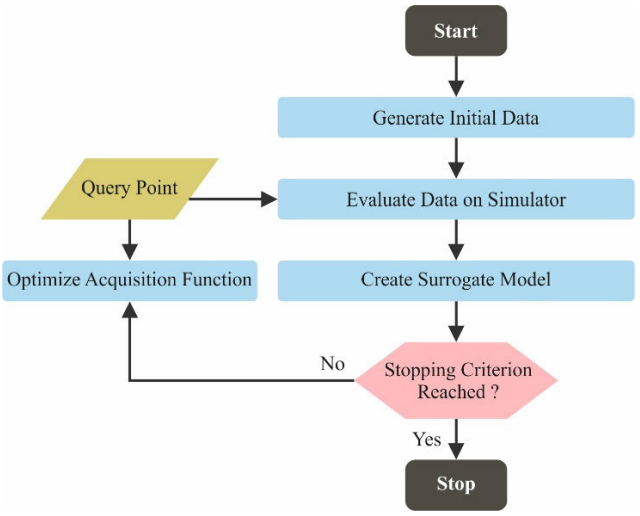

FIGURE 6. Workflow of BOA approach.

that need careful calibration to attain greater outcomes. Fig. 6 depicts the workflow of the BOA technique.

Hyperparameters are constant during training, increasing the model's accuracy while concurrently lowering memory usage and training time. According to the description of the problem, various simulations use numerous hyperparameters. There are no optimal hyperparameters that utilize each model. The term BO designates a model that can probably be applied consecutively to optimize the parameter of some black-box function $f(x)$. BO incorporates previous beliefs to estimate a responding surface function $f^{\wedge}(x)$, using $f^{\wedge}(x)$ to select the configuration xn for trying to evaluate $f(xn)$ through true $f(x)$, identify posterior trust over-assessment of performance $f(xn)$, and carry on the process in a serial way till stopping principles have been reached at for tuning the testing sample to achieve better parameters that collaborated for improved classification.

The Bayesian theorem makes the BO basis. To upgrade the optimizer's posterior function, it begins a previous optimization function and gathers information from the preceding set of samples. Eq. (2), which states for model A and observation B, represents a base after the optimization process that depends on Bayes' Theorem.

$$P(A \mid B) = \frac{(P(B \mid A)\, P(A))}{P(B)} \tag{31}$$

whereas $P(AIB)$ signifies the probability of A assumed B, $P(B \mid A)$ characterizes the probability of B assumed A, $P(A)$ specifies the previous likelihood of A, then $P(B)$ indicates the margin likelihood of B. BO has been applied to define the minimum function value on a restricted set.

The fitness selection is a significant factor in operating the BOA performance method. The hyperparameter selection process includes the solution encoder model to estimate the efficiency of the candidate solutions. During this study, the BOA method evaluates precision as the main pattern for

designing the fitness function that is formulated below:

$$Fitness = \max(P) \tag{32}$$

$$P = \frac{TP}{TP + FP} \tag{33}$$

As a result of the formulation, TP represents the true positive, and FP specifies the false positive values.

IV. PERFORMANCE ANALYSIS

In this sector, the performance validation of the AETL-CDLPS system is carried out using the Bot-IoT dataset [36]. The dataset contains 15000 samples under nine class labels, as demonstrated in Table 1. The total number of features is 45, and 26 features have been selected. 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. Details of the dataset.

Class	Labels	No. of Samples
Service Scanning	C-1	2000
OS Fingerprinting	C-2	1500
DDoS TCP	C-3	2000
DDoS UDP	C-4	2000
DDoS HTTP	C-5	1000
DoS TCP	C-6	2000
DoS UDP	C-7	2000
DoS HTTP	C-8	1500
Normal	C-9	1000
Total Samples		15000

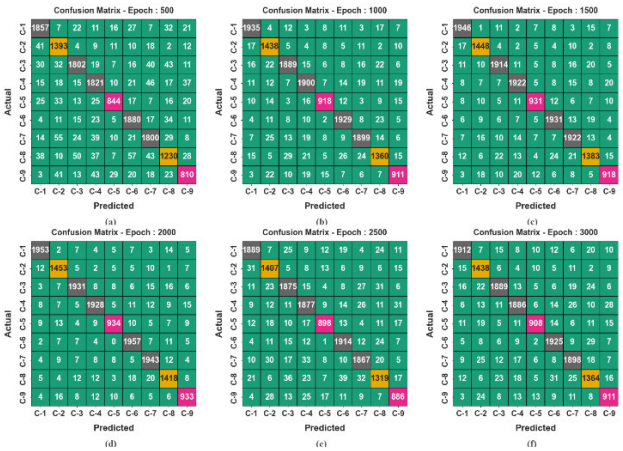

FIGURE 7. Confusion matrices of AETL-CDLPS technique (a-f) Epochs 500-3000.

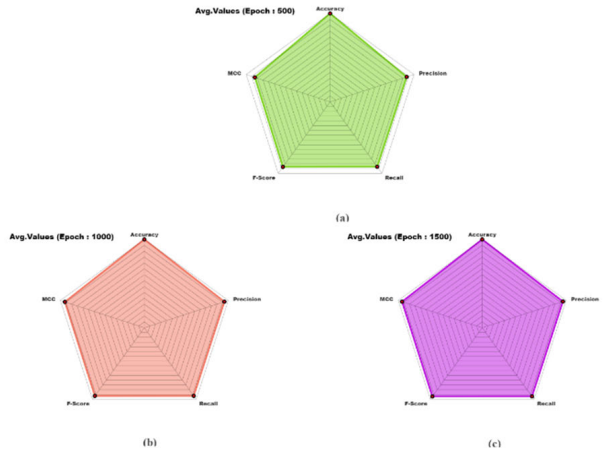

FIGURE 8. Average outcome of AETL-CDLPS algorithm (a-c) Epochs 500-1500.

Fig. 7 displayed the confusion matrices produced by the AETL-CDLPS technique under distinct epochs. The outcomes indicate that the AETL-CDLPS method has effectual recognition and classification of nine classes.

Table 2 represents the overall attack detection results of the AETL-CDLPS algorithm under various epochs and measures.

Fig. 8 illustrates the classification results of the AETL-CDLPS methodology under 500-1500 epochs. The resulting outcome implied that the AETL-CDLPS technique has correctly identified nine classes. With 500 epochs, the AETL-CDLPS system offers $accu_y$ of 97.68%, $prec_n$ of 89.14%, $reca_l$ of 88.70%, F_{score} of 88.88%, and MCC of 87.60%, respectively. Besides, with 1000 epochs, the AETL-CDLPS approach provides $accu_y$ of 98.78%, $prec_n$ of 94.23%, $reca_l$ of 94.11%, F_{score} of 94.17%, and MCC of 93.49%, correspondingly.

Fig. 9 shows the classifier outcomes of the AETL-CDLPS system under 2000-3000 epochs. The resulting outcome implied that the AETL-CDLPS model has correctly identified nine classes. With 2000 epochs, the AETL-CDLPS technique provides $accu_y$ of 99.19%, $prec_n$ of 96.10%, $reca_l$ of 95.97%, F_{score} of 96.03%, and MCC of 95.58%, correspondingly. Besides, with 3000 epochs, the AETL-CDLPS approach offers $accu_y$ of 98.71%, $prec_n$ of 93.87%, $reca_l$ of 93.80%, F_{score} of 93.82%, and MCC of 93.10%, respectively.

Fig. 10 illustrates the training and validation accuracy outcomes of the AETL-CDLPS system. The precision values are computed throughout 0-2000 epochs. The figure emphasizes that the training and validation accuracy values show a rising trend, which indicates the proficiency of the AETL-CDLPS method with improved performance over several iterations. In addition, the training and validation accuracy remain closer under the epochs, indicating low minimal overfitting and exhibiting the AETL-CDLPS approach's enhanced outcome, guaranteeing consistent prediction on unseen samples.

Fig. 11 shows the AETL-CDLPS system's training and validation loss graph. The loss values are computed for 0-2000 epochs. The training and validation accuracy values show a decreasing trend, which indicates the method's capability of balancing a trade-off between data fitting and generalization.

TABLE 2. Attack detection of AETL-CDLPS algorithm under various epochs.

Class Labels	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
Epoch - 500					
C-1	97.91	91.61	92.85	92.23	91.03
C-2	97.91	87.06	92.87	89.87	88.76
C-3	97.64	92.03	90.10	91.06	89.70
C-4	97.43	89.84	91.05	90.44	88.96
C-5	98.33	89.88	84.40	87.06	86.21
C-6	97.94	90.87	94.00	92.41	91.23
C-7	97.36	90.18	90.00	90.09	88.57
C-8	96.89	86.26	82.00	84.07	82.39
C-9	97.75	84.55	81.00	82.74	81.55
Average	97.68	89.14	88.70	88.88	87.60
Epoch - 1000					
C-1	99.01	95.89	96.75	96.32	95.75
C-2	98.82	92.59	95.87	94.20	93.56
C-3	98.68	95.60	94.45	95.02	94.26
C-4	98.62	94.67	95.00	94.83	94.04
C-5	99.06	93.96	91.80	92.87	92.37
C-6	98.91	95.45	96.45	95.95	95.32
C-7	98.73	95.48	94.95	95.21	94.48
C-8	98.37	92.83	90.67	91.74	90.84
C-9	98.85	91.65	91.10	91.37	90.76
Average	98.78	94.23	94.11	94.17	93.49
Epoch - 1500					
C-1	99.16	96.43	97.30	96.86	96.38
C-2	99.14	94.95	96.53	95.74	95.26
C-3	98.93	96.23	95.70	95.96	95.35
C-4	98.95	96.00	96.10	96.05	95.44
C-5	99.21	94.90	93.10	93.99	93.57
C-6	99.03	96.17	96.55	96.36	95.80
C-7	98.87	95.43	96.10	95.76	95.11
C-8	98.63	93.95	92.20	93.07	92.31
C-9	98.97	92.63	91.80	92.21	91.66
Average	98.99	95.19	95.04	95.11	94.54
Epoch - 2000					
C-1	99.37	97.65	97.65	97.65	97.29
C-2	99.25	95.72	96.87	96.29	95.88
C-3	99.17	97.23	96.55	96.89	96.41
C-4	99.13	97.03	96.40	96.71	96.21
C-5	99.27	95.50	93.40	94.44	94.05
C-6	99.26	96.64	97.85	97.24	96.82
C-7	99.11	96.19	97.15	96.67	96.15
C-8	98.95	94.91	94.53	94.72	94.14
C-9	99.16	94.05	93.30	93.67	93.23
Average	99.19	96.10	95.97	96.03	95.58
Epoch - 2500					
C-1	98.58	94.88	94.45	94.66	93.84
C-2	98.48	91.25	93.80	92.50	91.67
C-3	98.29	93.42	93.75	93.59	92.60
C-4	98.23	92.97	93.85	93.41	92.39
C-5	98.85	92.67	89.80	91.21	90.61
C-6	98.63	94.10	95.70	94.89	94.10
C-7	98.29	93.82	93.35	93.58	92.60
C-8	97.90	90.78	87.93	89.33	88.18
C-9	98.51	89.05	88.60	88.82	88.03
Average	98.42	92.55	92.36	92.45	91.56
Epoch - 3000					
C-1	98.90	96.13	95.60	95.86	95.23
C-2	98.76	92.06	95.87	93.93	93.26
C-3	98.69	95.65	94.45	95.04	94.29
C-4	98.62	95.30	94.30	94.80	94.00
C-5	99.01	94.09	90.80	92.42	91.90
C-6	98.84	95.11	96.25	95.68	95.01
C-7	98.57	94.38	94.90	94.64	93.81
C-8	98.28	91.79	90.93	91.36	90.41
C-9	98.75	90.29	91.10	90.69	90.02
Average	98.71	93.87	93.80	93.82	93.10

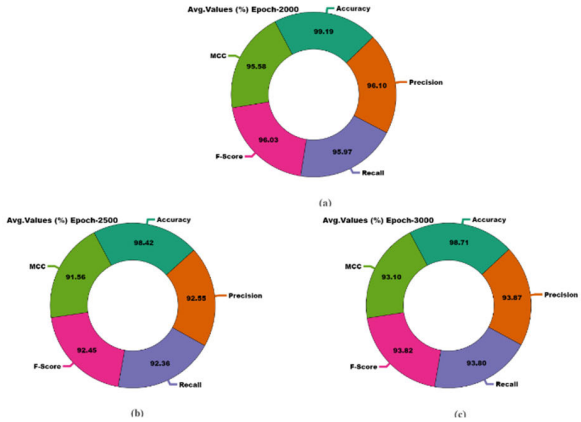

FIGURE 9. Average outcome of AETL-CDLPS algorithm (a-c) Epochs 2000-3000.

The continual reduction in loss values additionally guarantees the enhanced performance of the AETL-CDLPS technique and tunes the prediction outcomes over time.

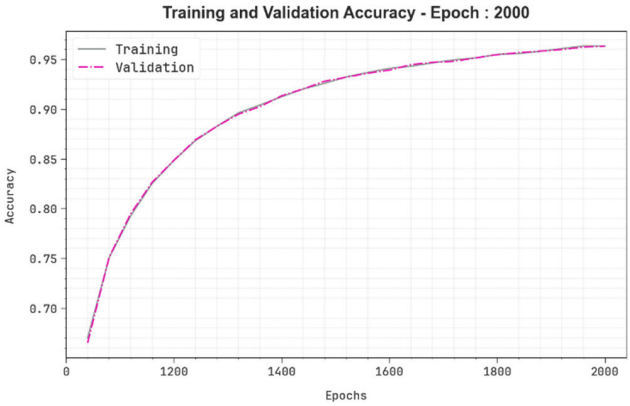

FIGURE 10. $Accu_y$ curve of AETL-CDLPS model under 2000 epochs.

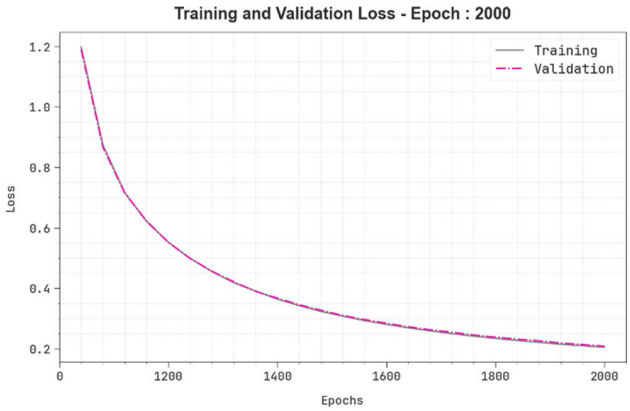

FIGURE 11. Loss curve of AETL-CDLPS model under 2000 epochs.

In Fig. 12, the precision-recall (PR) curve study of the AETL-CDLPS model interprets its performance by plotting Accuracy against Recall for all the classes. The figure

displays that the AETL-CDLPS system continuously accomplishes improved PR values across different class labels, indicating its capability to maintain a significant portion of true positive predictions amongst each positive prediction (precision) while also capturing a large proportion of actual positives (recall). The steady rise in PR results among all classes portrays the effectiveness of the AETL-CDLPS process in the classification process.

In Fig. 13, the ROC investigation of the AETL-CDLPS process is studied. The outcomes imply that the AETL-CDLPS system reaches enhanced ROC outcomes over each class, displaying a significant ability to discriminate the classes. This reliable tendency of improved ROC values over various classes signifies the efficient performance of the AETL-CDLPS approach in predicting classes, emphasizing the robust nature of the classification process.

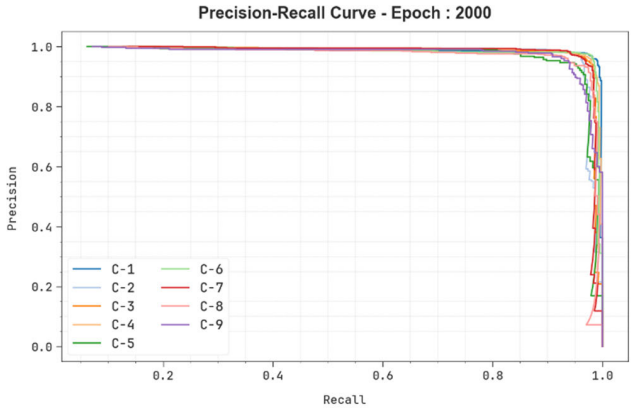

FIGURE 12. PR curve of AETL-CDLPS model under 2000 epochs.

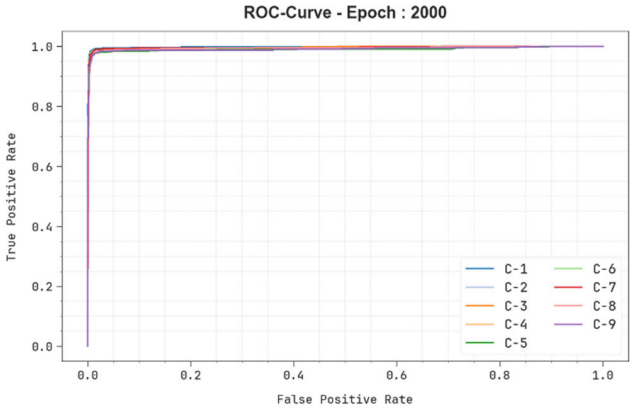

FIGURE 13. ROC curve of AETL-CDLPS method under 2000 epochs.

Table 3 and Fig. 14 show the AETL-CDLPS system's experimental results with recent models [37], [38], [39]. The results show that the ensemble bag model has shown worse performance with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 92.47%, 92.04%, 91.32%, and 93.06%, respectively. At the same time, the SVM method has attained slightly increased results with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 92.66%, 91.14%, 91.58%, and 93.16%, respectively. Besides, the DT, sparse AE, CNN-BiLSTM, and CANET techniques have achieved moderately closer performance. Meanwhile, the FNN-Focal model has resulted in considerable outcomes with $accu_y$, $prec_n$, $reca_l$, and F_{score} of 98.25%, 95.70%, 94.81%, and 95.69%, respectively. But the AETL-CDLPS process outperforms the other models with maximum $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.19%, 96.10%, 95.97%, and 96.03%, respectively.

TABLE 3. Comparative analysis of the AETL-CDLPS model with recent models.

Model	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
Ensemble Bag	92.47	92.04	91.32	93.06
SVM Classifier	92.66	92.14	91.58	93.16
Decision Tree	93.65	92.26	91.74	93.29
SSAE	96.09	92.65	93.27	95.07
CNN-BiLSTM	96.10	94.60	94.30	95.21
CANET	96.65	94.89	94.77	95.22
FNN-Focal	98.25	95.70	94.81	95.69
AETL-CDLPS	99.19	96.10	95.97	96.03

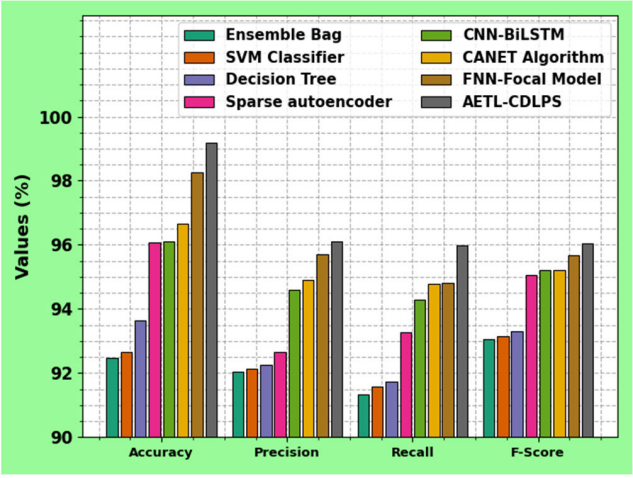

FIGURE 14. Comparative analysis of the AETL-CDLPS technique with recent models.

The computational time (CT) of the AETL-CDLPS process can be associated with existing models in Table 4 and Fig. 15. The results emphasized that the SVM and CANET approaches have attained the least performance with increased CT of 9.61s and 8.25s, correspondingly. The CNN-BiLSTM, DT, FNN-Focal, and sparse AE techniques have reported closer CT values of 7.48s, 6.86s, 5.98s, and 4.44s, respectively. Meanwhile, the ensemble bag model has managed to report a considerable CT of 4.14s. Nevertheless, the AETL-CDLPS method exhibited superior performance with a minimal CT of 2.75s. The AETL-CDLPS approach exhibits a lesser CT of 2.75 seconds due to its optimized algorithmic design, which streamlines data processing and mitigates overhead. The methodology accelerates both the training and inference stages by employing effectual feature

selection and minimizing complex calculations. Moreover, its architecture is likely constructed to handle input data efficiently, enabling faster convergence and real-time performance in cyberattack detection and classification tasks. Therefore, the AETL-CDLPS approach can be employed for accurate cyberattack detection and classification processes.

TABLE 4. CT analysis of AETL-CDLPS technique with existing models.

Model	CT (sec)
Ensemble Bag	4.14
SVM Classifier	9.61
Decision Tree	6.86
SSAE	4.44
CNN-BiLSTM	7.48
CANET	8.25
FNN-Focal	5.98
AETL-CDLPS	2.75

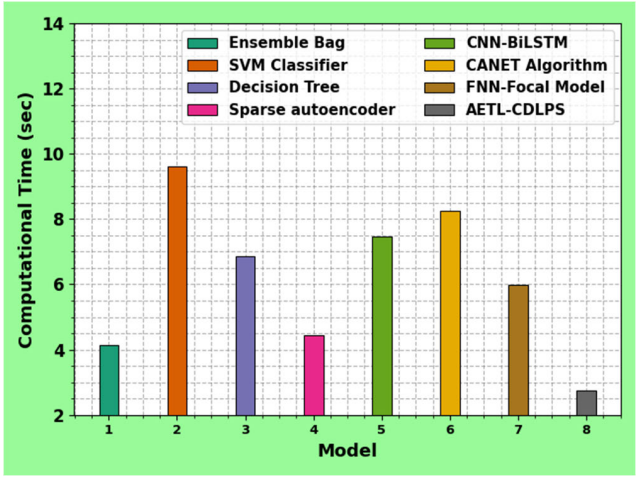

FIGURE 15. CT analysis of the AETL-CDLPS technique with recent models.

V. CONCLUSION

In this work, we concentrate on the design of the AETL-CDLPS technique. The major aim of the AETL-CDLPS technique lies in the automated detection of cyber-attacks for IoT-assisted resource-constrained systems. To achieve this, the AETL-CDLPS technique encompasses various stages, such as data preprocessing, feature selection, knowledge transfer through ensemble transfer learning, and hyperparameter tuning. Initially, the AETL-CDLPS technique utilizes LSN to normalize the input data. Next, the AETL-CDLPS technique employs an ICOA-based feature selection technique to choose optimal features. For the cyberthreat detection process, an ensemble transfer learning model comprises three classifiers: DCNN, GRU, and SSAE. Finally, the BOA method is used to optimize the hyperparameter tuning of the three ensemble techniques. The AETL-CDLPS model's performance validation is performed using the Bot-IoT dataset. The comparison study of the AETL-CDLPS method portrayed superior Accuracy, Precision, Recall, and F-Score values of 99.19%, 96.10%, 95.97%, and 96.03% over existing models. The limitations of the proposed method encompass its dependance on intrinsic models that may not perform optimally in resource-constrained environments, potentially affecting real-time detection capabilities. Furthermore, the framework may need help with scalability as the number of connected devices increases, leading to challenges in maintaining performance. Future work should focus on improving the effectualness and adaptability of the model to diverse environments, exploring lightweight alternatives that retain high accuracy. Moreover, incorporating growing AI models and addressing the practical threats of deployment in real-world scenarios, specifically in healthcare and smart home applications, could substantially enhance the model's efficiency. Lastly, further research should examine methods for balancing detection accuracy with computational resource needs.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/86/45. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R361), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2024R838), King Saud University, Riyadh, Saudi Arabia. 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-2024-2899-01. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

REFERENCES

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

[2] Z. Zhi-Xian and F. Zhang, "Image real-time detection using LSE-YOLO neural network in artificial intelligence-based Internet of Things for smart cities and smart homes," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–8, Mar. 2022.

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

[4] N. Torres, P. Pinto, and S. I. Lopes, "Security vulnerabilities in LPWANs-An attack vector analysis for the IoT ecosystem," *Appl. Sci.*, vol. 11, no. 7, p. 3176, 2021.

[5] W. B. Arfi, I. B. Nasr, T. Khvatova, and Y. B. Zaied, "Understanding acceptance of eHealthcare by IoT natives and IoT immigrants: An integrated model of UTAUT, perceived risk, and financial cost," *Technol. Forecast. Soc. Change*, vol. 163, Feb. 2021, Art. no. 120437.

[6] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Jun. 2018.

[7] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020.
[8] U. Khurana, H. Samulowitz, and D. Turaga, "Feature engineering for predictive modeling using reinforcement learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Apr. 2018.
[9] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli, and E. Cambiaso, "Exploiting Internet of Things protocols for malicious data exfiltration activities," *IEEE Access*, vol. 9, pp. 104261–104280, 2021.
[10] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable intrusion detection for cyber defences in the Internet of Things: Opportunities and solutions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1775–1807, 3rd Quart.,3, 2023.
[11] M. M. Salim, D. Camacho, and J. H. Park, "Digital twin and federated learning enabled cyberthreat detection system for IoT networks," *Future Gener. Comput. Syst.*, vol. 161, pp. 701–713, Dec. 2024.
[12] S. AboulEla, N. Ibrahim, S. Shehmir, A. Yadav, and R. Kashef, "Navigating the cyber threat landscape: An in-depth analysis of attack detection within IoT ecosystems," *AI*, vol. 5, no. 2, pp. 704–732, May 2024.
[13] S. Bergies, T. M. Aljohani, S.-F. Su, and M. Elsisi, "An IoT-based deep-learning architecture to secure automated electric vehicles against cyberattacks and data loss," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 54, no. 9, pp. 5717–5732, Sep. 2024.
[14] C. Rookard and A. Khojandi, "RRIoT: Recurrent reinforcement learning for cyber threat detection on IoT devices," *Comput. Secur.*, vol. 140, May 2024, Art. no. 103786.
[15] P. Rana, S. Chauhan, and B. P. Patil, "Cyber security threats detection in IoT using krill based deep neural network stacked auto encoders," *Wireless Pers. Commun.*, vol. 135, no. 1, pp. 299–322, Mar. 2024.
[16] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 110–128, Jan. 2024.
[17] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in industrial Internet of Things," *Digit. Commun. Netw.*, vol. 9, no. 1, pp. 101–110, Feb. 2023.
[18] F. Alrowais, S. Althahabi, S. S. Alotaibi, A. Mohamed, M. A. Hamza, and R. Marzouk, "Automated machine learning enabled cybersecurity threat detection in Internet of Things environment," *Comput. Syst. Sci. Eng.*, vol. 45, no. 1, pp. 687–700, 2023.
[19] L. Sana, M. M. Nazir, J. Yang, L. Hussain, Y.-L. Chen, C. S. Ku, M. Alatiyyah, S. A. Alateyah, and L. Y. Por, "Securing the IoT cyber environment: Enhancing intrusion anomaly detection with vision transformers," *IEEE Access*, vol. 12, pp. 82443–82468, 2024.
[20] H. Wang, H. Zhang, L. Zhu, Y. Wang, and J. Deng, "ResADM: A transfer-learning-based attack detection method for cyber–physical systems," *Appl. Sci.*, vol. 13, no. 24, p. 13019, Dec. 2023.
[21] I. Ioannou, P. Nagaradjane, P. Angin, P. Balasubramanian, K. J. Kavitha, P. Murugan, and V. Vassiliou, "GEMLIDS-MIOT: A green effective machine learning intrusion detection system based on federated learning for medical IoT network security hardening," *Comput. Commun.*, vol. 218, pp. 209–239, Mar. 2024.
[22] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications," *Sensors*, vol. 23, no. 11, p. 5348, Jun. 2023.
[23] S. Jaradat, M. M. Komol, M. Elhenawy, and N. Dong, "Cyber attack detection on SWaT plant industrial control systems using machine learning," *Artif. Intell. Auto. Syst.*, 2024.
[24] M. R. Willeke, D. A. Bierbrauer, and N. D. Bastian, "Data-efficient, federated learning for raw network traffic detection," *Proc. SPIE*, vol. 12538, pp. 247–262, Jun. 2023.
[25] H. Asgharzadeh, A. Ghaffari, M. Masdari, and F. S. Gharehchopogh, "An intrusion detection system on the Internet of Things using deep learning and multi-objective enhanced gorilla troops optimizer," *J. Bionic Eng.*, vol. 21, no. 5, pp. 2658–2684, Sep. 2024.
[26] D. Javeed, T. Gao, M. S. Saeed, P. Kumar, R. Kumar, and A. Jolfaei, "A softwarized intrusion detection system for IoT-enabled smart healthcare system," *ACM Trans. Internet Technol.*, Nov. 2023.
[27] A. Tiwari and A. A. Waoo, "IoT based smart home cyber-attack detection and defense," *TIJER-Int. Res. J.*, vol. 10, no. 8, 2023.
[28] U. Zukaib, X. Cui, C. Zheng, M. Hassan, and Z. Shen, "Meta-IDS: Meta-learning-based smart intrusion detection system for Internet of Medical Things (IoMT) network," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23080–23095, Jul. 2024.
[29] R. Verma and S. Chandra, "RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu," *Eng. Appl. Artif. Intell.*, vol. 118, Feb. 2023, Art. no. 105670.
[30] A. M. Nevill and R. L. Holder, "Scaling, normalizing, and per ratio standards: An allometric modeling approach," *J. Appl. Physiol.*, vol. 79, no. 3, pp. 1027–1031, Sep. 1995.
[31] K. Ji, A. Dogani, N. Jin, and X. Zhang, "Integrating improved coati optimization algorithm and bidirectional long short-term memory network for advanced fault warning in industrial systems," *Processes*, vol. 12, no. 3, p. 479, Feb. 2024.
[32] N. Waddenkery and S. Soma, "Loitering based human crime detection in video surveillance using Beluga whale Adam dingo optimizer and deep convolutional neural network," *Int. Arab J. Inf. Technol.*, vol. 21, no. 3, 2024.
[33] M. Cheng, Q. Zhang, and Y. Cao, "An early warning model for turbine intermediate-stage flux failure based on an improved HEOA algorithm optimizing DMSE-GRU model," *Energies*, vol. 17, no. 15, p. 3629, Jul. 2024.
[34] T. Jorge, J. Magalhães, R. Silva, A. Guedes, D. Ribeiro, C. Vale, A. Meixedo, A. Mosleh, P. Montenegro, and A. Cury, "Early identification of out-of-roundness damage wheels in railway freight vehicles using a wayside system and a stacked sparse autoencoder," *Vehicle Syst. Dyn.*, pp. 1–26, Mar. 2024.
[35] A. M. Elshewey, M. Y. Shams, N. El-Rashidy, A. M. Elhady, S. M. Shohieb, and Z. Tarek, "Bayesian optimization with support vector machine model for Parkinson disease classification," *Sensors*, vol. 23, no. 4, p. 2085, Feb. 2023.
[36] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.
[37] S. Alosaimi and S. M. Almutairi, "An intrusion detection system using BoT-IoT," *Appl. Sci.*, vol. 13, no. 9, p. 5427, Apr. 2023.
[38] X. Liu and Y. Du, "Towards effective feature selection for IoT botnet attack detection using a genetic algorithm," *Electronics*, vol. 12, no. 5, p. 1260, Mar. 2023.
[39] B. Xu, L. Sun, X. Mao, R. Ding, and C. Liu, "IoT intrusion detection system based on machine learning," *Electronics*, vol. 12, no. 20, p. 4289, Oct. 2023.