Journal of Intelligent Systems and Internet of Things
Vol. 16, No. 01, PP. 118-131, 2025
DOI: https://doi.org/10.54216/JISIoT.160110

Robust Zero-Day Attack Detection with Optimal Deep Learning for Securing Internet of Things Environment

Nahla J. Abid[1], Nawaf Alhebaishi[2,*], Turki Althaqafi[3]

[1]Department of Computer Science, Taibah University, Madinah, Saudi Arabia

[2]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

[3]Computer Science Department, School of Engineering, Computing and Design, Dar Al-Hekma University, Jeddah, Saudi Arabia

Emails: nabd@taibahu.edu.sa; nalhebaishi@kau.edu.sa; tthaqafi@dah.edu.sa

Abstract

The Internet of Things (IoT) aims to provide connectivity between all computing entities. However, this facilitates cyberthreats, which exploits the existence of vulnerability over a period. The zero-day threat is one of the vulnerabilities that can result in zero-day attacks that are destructive to the network security and an enterprise. This attack may have potentially compromised critical infrastructure, far-reaching consequences, national security, and even personal privacy. To alleviate the risks, organizations and manufacturers should prioritize proactive security measures, involving robust authentication mechanisms, ongoing monitoring, and timely software updates, to defend the IoT ecosystem from emerging threats. In present scenario, deep learning (DL)-based models have improved robustness in learning data giving it an improved capability to identify unknown information, since it can able to extract knowledge of non-linear data to identify unknown information. The study presents a Robust Zero-Day Attack Detection with Optimal Deep Learning (RZDAD-ODL) technique for the IoT framework. The primary intention of the RZDAD-ODL model lies in the automatic and effectual detection of zero-day attacks in the IoT framework. In the presented RZDAD-ODL technique, the honey badger algorithm (HBA) can be used for the optimum range of the features. Besides, the RZDAD-ODL technique exploits the conditional variational autoencoder (CVAE) model for attack detection and its parameter tuning process can be performed by using a rider optimization algorithm (ROA). The experimentation results of the RZDAD-ODL system can be validated on a benchmark dataset. Extensive comparison studies reported the better attack detection performance of the RZDAD-ODL model over other current techniques.

Received: November 21, 2024 Revised: January 04, 2025 Accepted: February 12, 2025

Keywords: Internet of Things; Zero-day attacks; Deep learning; Feature selection; Cybersecurity

1. Introduction

In recent times, IoT network comprises a number of smart devices that interact with each other requiring less human interference. By the end of this year, it is expected that masses of IoT devices will increase in 2022, expecting 46 billion devices [1]. IoT technology has become vital in real-time smart applications within the smart cities and smart industry. Through the invention of smart applications such as industrial, medical, habitat education, etc, IoT has become the best solution. It extends the potential of the Internet beyond computers to a wide array of processes and environments [2]. For gathering and sending data back or both, the user can be monitored into the network. With this innovation, the Internet embedded over 99% of objects and environments that usually remain beyond the reach of the network. The wide application of IoT devices makes it fertile ground for malevolent users to perform cyberattacks [3].

In general, a zero-day attack is considered a "traffic pattern of interest without equivalent patterns in attack detection element or malware in the network" [4]. The authors highlight their prevalence and impact. The outcome shows that zero-day attacks can compromise the system and exist for a considerable amount of time (a normal of ten months) earlier they are identified [5]. Conventional cyber-security system protects devices and users via anti-virus software, user authentication, Intrusion Detection System (IDS), encryption of data, and firewalls [6]. As illustrated by Kwon et al., the usage of Machine Learning (ML) algorithms to identify abnormal activities, attempts in computer systems, and malicious network traffic in IDS is not sufficient [7]. Classical ML still lacks automated feature engineering; they are not effective in identifying slight variations of actual attacks and have a lower recognition rate. This has resulted in considering the DL technique to enhance cyber-security systems [8]. DL is a subfield of ML that has attracted wide attention in various fields due to its recent developments in hardware and software and improvement in accuracy in complex tasks. DL technique improves cyber-security systems to prevent attacks by recognizing patterns that are not similar to normal behaviour [9]. Cyberattack shares common features with image detection, over 99% of new attacks are lesser mutants of the present one; similarly, changes in images can be detected by slight variations in their pixels. The IoT-Fog network detects network attacks and threads. Although IoT features (viz., limited computational abilities and distributed nature of end-devices) requires new solution for the IDS [10].

The study presents a Robust Zero-Day Attack Detection with Optimal Deep Learning (RZDAD-ODL) technique for the IoT framework. The primary intention of the RZDAD-ODL model lies in the automatic and effective detection of zero-day attacks in the IoT framework. In the presented RZDAD-ODL technique, the honey badger algorithm (HBA) can be used for the optimum range of the features. Besides, the RZDAD-ODL technique exploits the conditional variational autoencoder (CVAE) model for attack detection and its parameter tuning process can be performed by using a rider optimization algorithm (ROA). The simulation analysis of the RZDAD-ODL system can be authorized on a benchmark dataset.

2. Related Works

Popoola et al. [11] devise the federated DL (FDL) technique for the recognition of zero-day attacks to avoid leakage of information in IoT devices. In the proposed approach, an optimum DNN algorithm is exploited for the classification of traffic. After several communication rounds between IoT devices and the model parameter server, a global DNN algorithm is created. Bu and Cho [12] introduce the incorporation of convolution operation to model a deep CAE (DCAE) and the character-level URL features to consider the basic idea of a zero-day attack. In [13], aims to authenticate the reliability of the detection method once it meets an attack that it was not previously trained. Thus, the classifier accuracy of CNN is evaluated for detecting the malicious attack.

With the fusion of Teaching Learning, based optimizer (TLBO) and Simulated Annealing (SA) (TLBOSA), Shukla [14] suggests a new hybrid mechanism for the IDS that extracts relevant attributes and removes the inappropriate features from the complex data. In the proposed method, SVM is exploited as a fitness function (FF) for selecting the important features that help to accurately categorize the attacks. Cheng et al. [15] present a cyber-situation perception ontology construction model was first provided to describe the activities of APT attacks. Lastly, a ZDAARA within APT (ZDAARA) is introduced to detect malicious events that could not be identified by the IDS. In [16], adopted an ensemble learning approach: a fusion of dissimilar base anomaly detectors constructed by classical ML algorithm to address the problems. The learning algorithm provides an accurate zero-day container recognition. Firstly, a testbed is constructed to enable data storage and collection, model inference and training.

In [17], an AE model for identifying the zero-day attack. The study focuses on constructing IDS with a high recognition rate whilst retaining a false negative rate. Two typical IDS data are applied for the NSL-KDD and CICIDS2017 evaluation. To determine the model efficacy, we compare its outcomes against a modern single-class SVM. In [18], introduced a Fuzzy Gaussian Mixture-based Correntropy (FGMC-HADS) by using the Gaussian mixture model (GMM), fuzzy rough set attribute reduction (FRAR) technique and Correntropy models. The FGMC-HADS includes two different modules: (i) the FRAR technique is used for combining the elapsed times and system calls' identifiers to build applicable hidden patterns; and (ii) the Correntropy and GMM mechanism is an anomaly detection method known as 'Corr-GMM', to combine multi-variate feature and identify unknown anomalous activities, correspondingly.

3. The Proposed model

In this manuscript, we have presented the RZDAD-ODL algorithm for the IoT framework. The primary goal of the RZDAD-ODL system lies in the automatic and effectual recognition of zero-day attacks in the IoT framework. The RZDAD-ODL technique consists of three different processes such as ROA-based parameter tuning, CVAE-based identification, and HBA-based feature extraction. Fig. 1 portrays the complete workflow of the RZDAD-ODL model.

A. Feature extraction using the HBA model

The HBA model is used in this stage for the optimum selection of features. HBA stimulates the feeding behavior of honey badger (HB) [19]. They have two ways to search for food while foraging. First, they use good olfactory senses to search and tactic the honey. Next, discover a better position near the nourishment sources. This phenomenon is called “digging mode”. HBs find honey by mimicking honeyguide birds is called “honey mode”. The arithmetical formula of HBA stimulates the foraging behaviours of HBs. Since HBA has exploitation and exploration stages, it is considered a global optimization approach.

Consider that the HBA enhances D-dimension solutions, which is defined as a population of the candidate solution:

$$N = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1D} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2D} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nD} \end{bmatrix} \quad (1)$$

In Eq. (1), n denotes the agents count and the location of the ith agent is $x_i = [x_i^1, x_i^2, \dots, x_i^D]$, the upper and lower limitations are correspondingly denoted by ub_i and $1b_i$.

The initial position of the agent is characterized by the following expression:

$$x_i = lb_i + r \times (ub_i - lb) \quad (2)$$

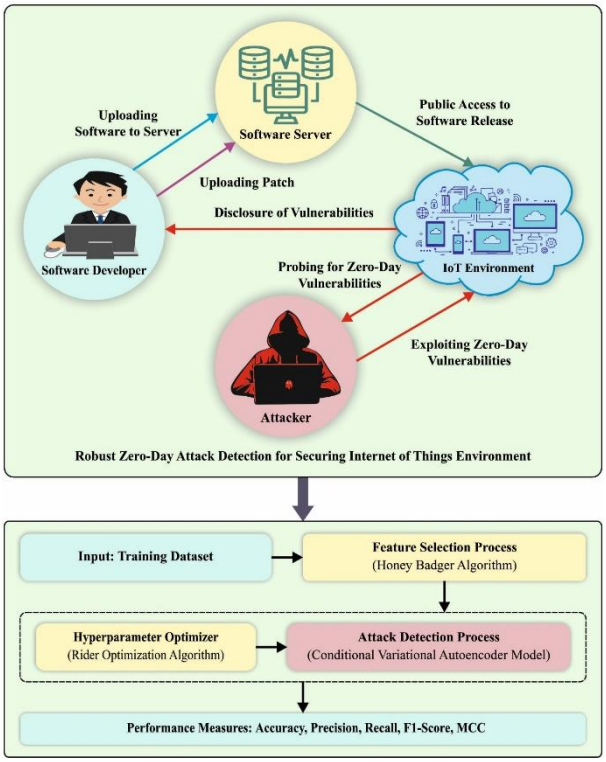

Figure 1. Overall flow of RZDAD-ODL algorithm

In Eq. (2), randomly generated numbers inside [0,1] is r.

:

$$I_i = r_1 \times \frac{S}{4\pi d_i^2} \quad (3)$$

In Eq. (3), a random value from zero to one is $r1$. The source strength is represented as S:

$$S = (x_i - x_{i-1})^2 \quad (4)$$

In Eq. (4), the distance between i^{th} HBs and prey is d_i:

$$d_i = x_{prey} - x_i \tag{5}$$

In Eq. (5), the location of prey detected as the optimum solution is x_{prey}. Determine the variable α. To enable a progressive shift from exploration to exploitation, variable α is represented as follows:

$$\alpha = C \times e^{\frac{-r}{t_{\max}}} \tag{6}$$

In Eq. (6), the maximal amount of iterations is $t_{\max}$, and C is a constant.

Once the HB iteratively updates the location, there are two models.

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_2 \times \alpha \times d_i \times |\cos(2\pi r_3) \times [1 - \cos(2\pi r_4)]| \tag{7}$$

Here fag F alters the search direction as follows:

$$F = \begin{cases} 1 \; if \; r_5 \leq 0.5 \\ -1 \; else \end{cases} \tag{8}$$

In Eq. (8), random numbers between 0 and 1 within r_2, r_3, r_4, and r_5, correspondingly. The constant β shows the capability of HBs to attain food. During the digging model, the behavioral patterns of HBs are similar to the structure of cardioid shape.

The new location of HB through iteration is defined in the honey model as follows:

$$x_{new} = x_{prey} + F \times r_6 \times \alpha \times d_i \tag{9}$$

In Eq. (9), a random integer within [0,1] is r_6.

The F permits the agent to shift the exploration direction to rise the probability of escape from the local optimal and comprehensively examining the searching space.

In the presented HBA model, the FF is intended to have a balance among the amount of features nominated (minimum) and the classifier accuracy (maximum). Eq. (10) displays the FF to assess the solution.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|C|} \tag{10}$$

Where the classifier rate of error is characterized as $\gamma_R(D)$. The cardinality of the selected subset is represented by $|R|$ and the overall quantity of attributes in the database is $|C|$, $\in$ [1,0] and $\beta = 1 - \alpha$, the variables α and β are resemble to the implication of classifier quality and subset length.

B. CVAE classification Model

For attack detection, we employed of CVAE model. Like AE, an important generative model, VAE has a similar network structure that includes encoding and decoding parts [20]. In the AE, the encoding part is used to define the mapping from input dataset $x \in \mathbb{R}^{d_x}$ to a hidden parameter $z \in \mathbb{R}^{d_z}$, whereas the decoding part is used to define mapping back from the hidden parameter z into input space. The AE makes the reconstructed X nearly the original one x and also learns hidden characteristics of typical data. In the VAE, the hidden parameter z is constrained to be distributed based on $p_\theta(z)$ prior distribution. The multi-variate unit Gaussian $\mathcal{N}(0, I)$, makes the CVAE system for learning an input dataset. While mapping from input dataset x to hidden parameter z, based on Eq. (11), $p_\theta(z|x)$ and $p_\theta(x)$ are intractable.

$$p_\theta(z|x) = \frac{p_\theta(x, z)}{p_\theta(x)} \tag{11}$$

Therefore, the Variational Inference technique is used to resolve these problems in a tractable way by discovering $q_\phi(z|x)$ approximation posterior.

$$q_\phi(z|x) = N(\mu_z, \sigma_z^2 I) \tag{12}$$

In Eq. (12), By using the encoder, the mean μ_z and standard deviation σ_z of approximation posterior $q_\phi(z|x)$ are derived.

Assume an inference model $q_\phi(z|x)$, the ELBO is derived by the following expression:

$$log p_\theta(x) = E_{q_\phi(z|x)}[\log p_\theta(x)] \tag{13}$$

$$= E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}\right] \tag{14}$$

$$= E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)} \frac{q_\phi(z|x)}{q_\phi(z|x)}\right] \tag{15}$$

$$= E_{q_\phi(z|x)}\left[\log p_\theta(xz) log p_\theta(z) log q_\varphi(zx)\right] + D_{KL}\left(q_\phi(z|x)||p_\theta(z|x)\right) \tag{16}$$

In Eq. (16), the ELBO (Evidence Lower Bound) is the initial term and the KL divergence of approximate $q_\phi(z|x)$ from the posterior $p_\theta(z|x)$ is the next term. The KL divergence between them should be reduced to ensure $q_\phi(z|x)$ gets closer to $p_\theta(z|x)$. Based on the equation, minimalizing KL divergence is transmuted into the tasks of maximizing ELBO:

$$\mathcal{L}_{VAE}(\theta, \emptyset; x) = -E_{q_\phi(z|x)}\left[\log p_\theta(x|z) + \log p_\theta(z) — log q_\phi(z|x)\right] \tag{17}$$

VAE is effectively used in various fields. Using a sliding window, VAE realizes anomaly detection in time sequence data. However, a typical VAE with a sliding window can deal with univariate time series data.

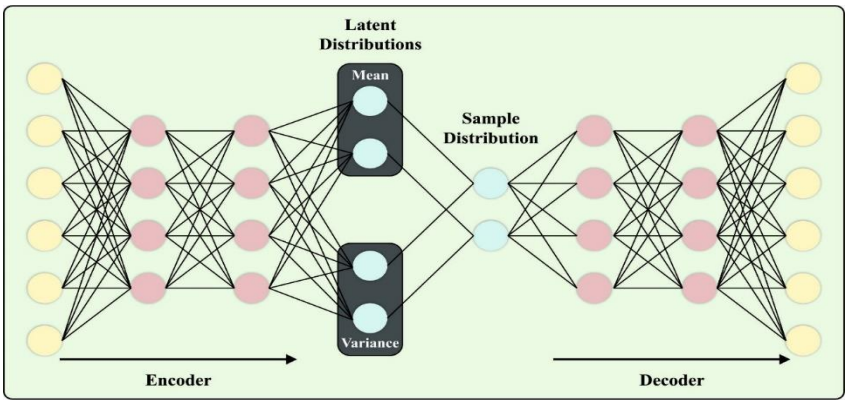

Figure 2. Architecture of CVAE

As a generalization model, CVAE encodes the input information into hidden space with Gaussian distribution. Next, the sample is decoded from the hidden space into an appropriate form [21]. The trained VAE can be navigated by producing new information and is continuous. The decoder module is used for generating the new sample. The input is given a random sample and emotion label with the size of hidden space. Fig. 2 demonstrates the infrastructure of CVAE.

The CVAE model includes encoding and decoding parts. The encoding part takes input x and evaluates the mean μ of multi-variate Gaussian distribution. The decoding part captures the sample from hidden vector z to recreate the input on output as $\tilde{x}$. The sum of Latent loss ($\mathcal{L}_L$) and Reconstruction loss ($\mathcal{L}_R$) can be represented by the loss function. Using cross-entropy, the reconstructed loss computes the distinction between input x and output $\tilde{x}$. Hidden can be evaluated by the KL divergence that evaluates the distance between the actual and Gaussian distribution in hidden vector z:

$$\mathcal{L}_L = -\frac{1}{2}\sum_{i=1}^{K}(1 + \log\sigma_i^2 - \sigma_i^2 - \mu_i^2) \tag{18}$$

In Eq. (18), the dimensionality of hidden vector z is K. Where, standard deviation and mean of i^{th} dimensions of hidden vector z are σ_i and μ_i.

C. Parameter tuning using ROA

Lastly, the ROA technique alters the parameters linked to the CVAE model. The ROA model is based on the race to the finishing line mid the riders group [22]. The “follower,” “overtaker,” “bypass rider,” and “attacker” are the four-rider groups with unique winning strategies used for updating the solution. The trailing rider goes in lockstep behind the leader. The bypass rider finds the direct route towards the destination. The top speed of the attacker is evaluated according to the relative positioning of riders with regard to the destination. The seven stages of the

ROA process are discussed in this section. The rider group and their limit: The group of riders in the population are chosen arbitrarily.

$$B_n = \{B_n(x,y)\}; 1 \le x \le E; 1 \le y \le F \tag{19}$$

Now a single instant at time is n, the total sum of riders is represented as E, the overall duration of the optimization problem is F, and the position of x^{th} riders represents B_n.

$$E = L + F + O + A \tag{20}$$

The parameters L, F, O and A are utilized in Eq. (20) to represent the overall count of riders correspondingly. We could gather the rider's "gear" "steering," "brake," and "accelerator," features after the team has been set up. The navigation angle at time n is represented by S.

$$S_n = \{S_{x,y}^n\}; 1 \le x \le E; 1 \le y \le F \tag{21}$$

Now, the triangulation angle of x^{th} vehicles is represented as $S_{x,y}^n$. The early navigation angle is evaluated at 0-th period:

$$S_{x,y} = \begin{cases} \theta_n; iffi = 1 \\ S_{x,y-1} + \phi; ify \neq 1 and N_{x,y-1} + \phi \le 360 \\ S_{x,y-1} + \phi \le 360; otherwise \end{cases} \tag{22}$$

The x^{th} position angle of the rider is denoted as θ_i, and the coordinate angle is represented as ϕ. To measure success rate: The rider's success rate can be measured by the distance after "initializing the group and rider parameters" as follows:

$$s_x = \frac{1}{\|B_x - J_Q\|} \tag{23}$$

In Eq. (23), J_Q denotes the target location, and B_x indicates the x^{th} rider location. To declare the success rider: the success rate to meet the following rider is critical. The rider is considered to have a "high success rate" and the succeeding rider if they travel a short distance between the initial and last destination.

1. Update the position of the bypass rider: The position updating is arbitrarily made to prevent the rider since the rider is excluding the common route.

$$B_{n+1}^M(x,y) = \delta\left[B_n(\eta,y) * \beta(y) + B_n(\xi,y) * [1-\beta(y)]\right] \tag{24}$$

In Eq. (24), the random integer ξ and η lie between 1 and E, the random number β and δ lies between 0 and 1.

$$B_{n+1}^N(x,cs) = B^J(J,cs) + \left[cos\left(S_{x,cs}^n\right) * B^J(J,cs) * u_x^n\right] \tag{25}$$

In Eq. (25), cs denotes the coordinate selector, and the foremost rider location is B^J.

2. Modify the position of the overtaker: The three building blocks in updating the overtaker sites are "relative success rate", "direction indicator," and "coordinate selector". The equation for the overtaker site being updated is shown below.

$$B_{n+1}^V(x,cs) = B_n(x,cs) + [di_n^k(x) * B^J(J,cs)] \tag{26}$$

3. Modify the site of the attacker: The assailant's location is established by looking at where the followers are, and the leader is considered as primary target.

$$B_{n+1}^M(x,y) = B^J(J,cs) + \left[cos\left(S_{x,cs}^n\right) * B^j(J,cs) * u_x^n\right] \tag{27}$$

In Eq. (27), S_x^{can} is represented by x^{th} riders of i^{th} coordinates and $B_{n+1}^M(x,y)$ denotes the position of the rider. Once the new leader is identified and the riders' position is updated, then the success rate might be re-evaluated.

Recover the rider constraint with a better solution: It is essential to modify the rider positions to attain the optimum solution.

End process: The procedure should be repeatedly done until the cut of time (Tof) has passed to define who wins the bike race.

The FF is the substantial factor influencing the accuracy of ROA. The hyperparameter range comprises the solution encoding method to measure the competence of the candidate solution. The ROA system reflects the accuracy as main standard for developing FF.

$$Fitness = \max(P) \tag{28}$$

$$P = \frac{TP}{TP + FP} \tag{29}$$

Where TP and FP are the true and false positive values.

4. Experimental validation

In this study, the experimentation outcome of the RZDAD-ODL model is verified on the benchmark dataset comprising 20000 samples with 10 classes as specified in Table1.

Table 1: Details of dataset

Classes	No. of Instances
"Backdoor"	2000
"DoS"	2000
"DDoS"	2000
"Injection"	2000
"MITM"	2000
"Scanning"	2000
"Ransomware"	2000
"Password"	2000
"XSS"	2000
"Normal"	2000
Total Instances	20000

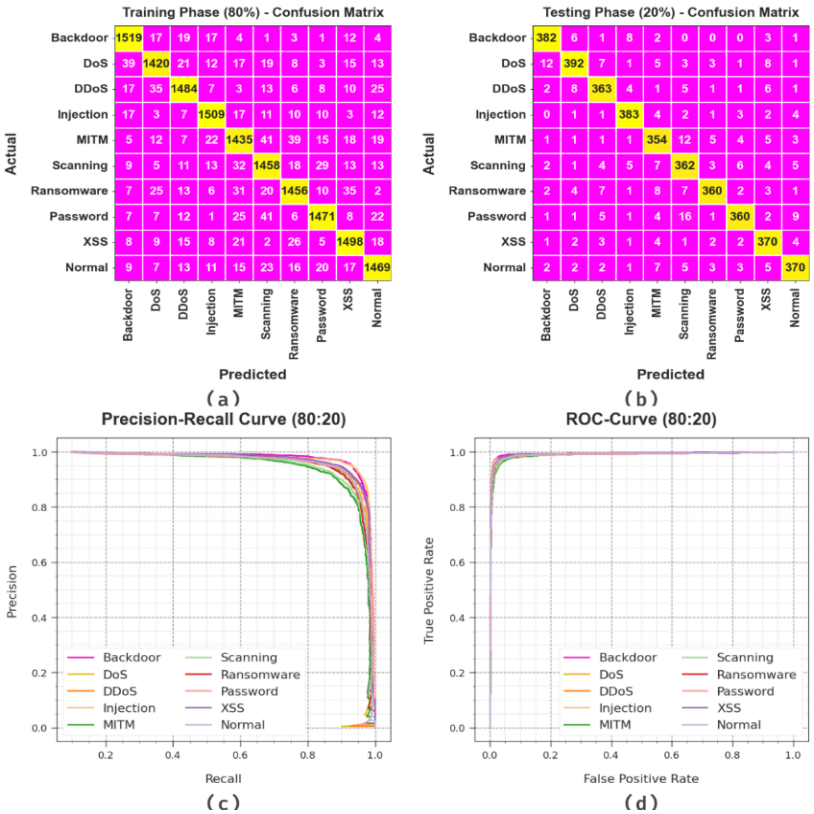

Figure 3. 80:20 of TRAS/TESS (a-b) Confusion matrices, (c) PR curve, and (d) ROC

Fig. 3 explains the classifier results of the RZDAD-ODL technique on 80:20 of TRAS/TESS. Figs. 3a-3b determines the confusion matrices offered by the RZDAD-ODL technique. The experimental result specified that the RZDAD-ODL system has discovered and recognized all 10 classes. In addition, Fig. 3c shows the PR result of the RZDAD-ODL system. The outcome specified that the RZDAD-ODL system has acquired the greatest PR outcome in 10 class labels. Lastly, Fig. 3d depicts the ROC curve of the RZDAD-ODL system. The outcome showed that the RZDAD-ODL algorithm resulted in high outcomes of ROC on 10 labels.

The zero-day attack recognition outcomes of the RZDAD-ODL algorithm are inspected on 80:20 of TRAS/TESS as portrayed in Fig. 4 and Table 2. The results highlighted that the RZDAD-ODL technique accurately detected all the classes. With 80% of TRAS, the RZDAD-ODL approach delivers an average $accu_y$ of 98.40%, $prec_n$ of 92%, $reca_l$ of 91.99%, F_{score} of 91.99%, and MCC of 91.11%. Likewise, with 20% of TESS, the RZDAD-ODL technique provides an average $accu_y$ of 98.48%, $prec_n$ of 92.43%, $reca_l$ of 92.41%, F_{score} of 92.40%, and MCC of 91.57%.

Table 2: Zero-day attack detection outcome of RZDAD-ODL approach with 80:20 of TRAS/TESS

Class	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	MCC
80% TRAS					
Backdoor	98.78	92.79	95.12	93.94	93.27
DoS	98.33	92.21	90.62	91.41	90.49
DDoS	98.49	92.63	92.29	92.46	91.62
Injection	98.83	93.96	94.37	94.17	93.52
MITM	97.86	89.69	88.96	89.32	88.13
Scanning	98.04	89.50	91.07	90.28	89.19
Ransomware	98.24	91.69	90.72	91.20	90.23
Password	98.56	93.58	91.94	92.75	91.96
XSS	98.48	91.96	93.04	92.50	91.65
Normal	98.38	91.98	91.81	91.90	91.00
Average	98.40	92.00	91.99	91.99	91.11
20% TESS					
Backdoor	98.90	94.32	94.79	94.55	93.94
DoS	98.32	93.78	90.53	92.13	91.21
DDoS	98.50	92.13	92.60	92.37	91.54
Injection	98.98	94.33	95.51	94.92	94.35
MITM	98.12	89.39	91.47	90.42	89.39
Scanning	97.80	87.65	90.73	89.16	87.95
Ransomware	98.65	94.99	91.14	93.02	92.30
Password	98.45	94.24	90.00	92.07	91.24
XSS	98.55	90.69	94.87	92.73	91.96
Normal	98.52	92.73	92.50	92.62	91.80
Average	98.48	92.43	92.41	92.40	91.57

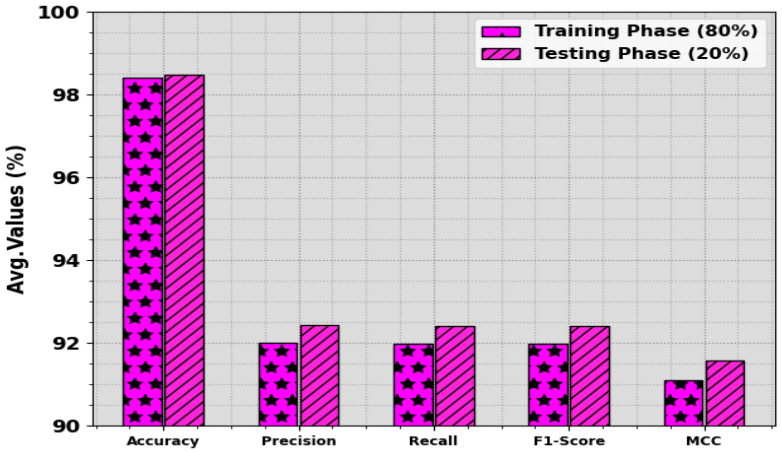

Figure 4. Average of RZDAD-ODL method with 80:20 of TRAS/TESS

Figure 5. $Accu_y$ curve of RZDAD-ODL approach with 80:20 of TRAS/TESS

As illustrated in Fig. 5, we have produced accuracy curves for the training (TRA) and testing (TES) data to calculate the efficiency of the RZDAD-ODL method with 80:20 of TRAS/TESS. This curve provides valuable insight into the model's learning development and its capacity to simplify. A notable development in both TRA and TES accuracy curves turn into marked as we upsurge the amount of epochs. This improvement signifies the ability of the model to well detect designs within the TRA and TES databases.

Fig. 6 shows a summary of the RZDAD-ODL model with 80:20 of TRAS/TESS, and the loss values of the model through the TRA model. The declining tendency in TRA loss over epochs specifies that the method continually perfect its weights to diminish forecast errors on TRA and TES datasets. This reveals how well the technique fits the TRA dataset. Notably, the TRA and TES loss consistently reduce, signifying the model's effectual learning of models current in both datasets. Additionally, it depicts the model's adaptation in lessening discrepancies among forecasts and the novel TRA classes.

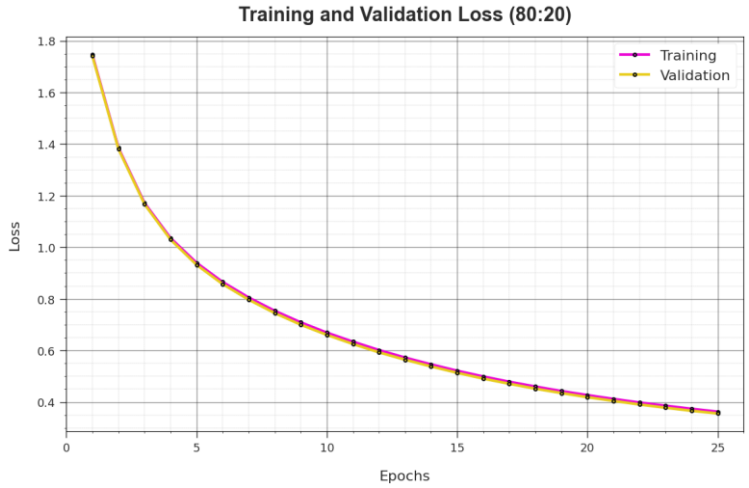

Figure 6. ROC curve of RZDAD-ODL approach with 80:20 of TRAS/TESS

Fig. 7 determines the classifier results of the RZDAD-ODL technique on 70:30 of TRAS/TESS. Figs. 7a-7b portrays the confusion matrices offered by the RZDAD-ODL approach. The experimental results denoted that the RZDAD-ODL system has distinguished and classified all 10 classes. Likewise, Fig. 7c shows the PR study of the RZDAD-ODL system. The result specified that the RZDAD-ODL technique has accomplished high PR outcomes in 10 classes. Besides, Fig. 7d determines the ROC examination of the RZDAD-ODL system. The result represented that the RZDAD-ODL method has leading to high values of ROC under 10 classes.

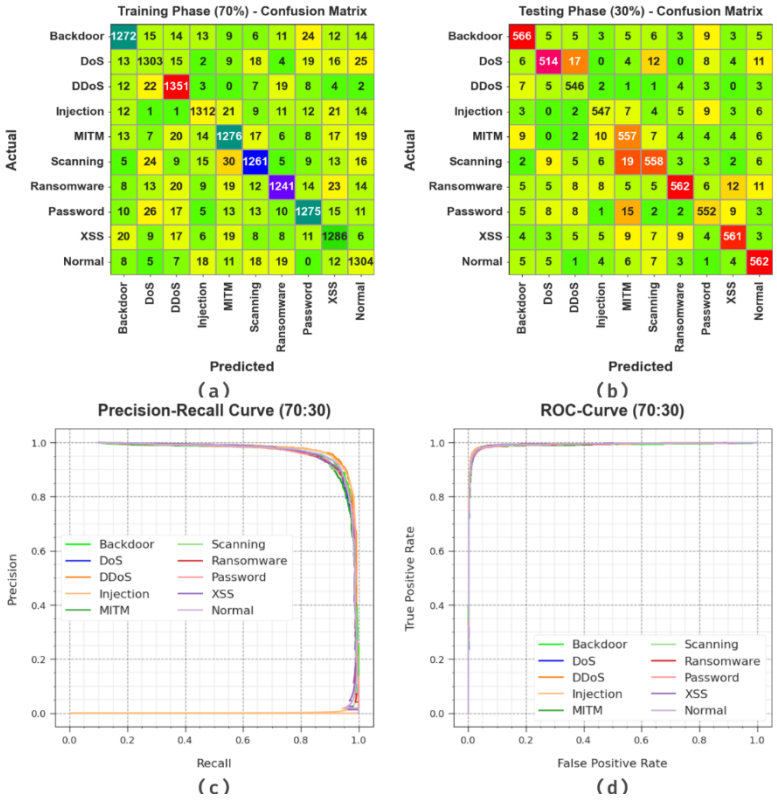

Figure 7. 70:30 of TRAS/TESS (a-b) Confusion matrices, (c) PR curve, and (d) ROC

The zero-day attack recognition results of the RZDAD-ODL method are inspected on 70:30 of TRAS/TESS and are portrayed in Table 3 and Fig. 8. The outcomes highlighted that the RZDAD-ODL method accurately detected all the classes. With 70% of TRAS, the RZDAD-ODL method provides an average $accu_y$ of 98.40%, $prec_n$ of 92.02%, $reca_l$ of 92%, F_{score} of 92%, and MCC of 91.12%. In addition, with 20% of TESS, the RZDAD-ODL system delivers an average $accu_y$ of 98.42%, $prec_n$ of 92.11%, $reca_l$ of 92.10%, F_{score} of 92.09%, and MCC of 91.22%.

Table 3: Zero-day attack detection outcome of RZDAD-ODL approach with 70:30 of TRAS/TESS

Classes	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	MCC
TRAS (70%)					
Backdoor	98.44	92.64	91.51	92.07	91.21
DoS	98.26	91.44	91.50	91.47	90.50
DDoS	98.59	91.84	94.61	93.20	92.43
Injection	98.66	93.92	92.79	93.35	92.61
MITM	98.20	90.69	91.34	91.01	90.01
Scanning	98.33	92.11	90.92	91.51	90.58
Ransomware	98.39	93.03	90.39	91.69	90.81
Password	98.39	92.39	91.40	91.89	91.00
XSS	98.31	90.63	92.52	91.56	90.63
Normal	98.44	91.51	93.01	92.25	91.39
Average	98.40	92.02	92.00	92.00	91.12
TESS (30%)					
Backdoor	98.50	92.48	92.79	92.64	91.80
DoS	98.30	92.78	89.24	90.97	90.06
DDoS	98.68	91.15	95.45	93.25	92.55
Injection	98.70	93.34	93.34	93.34	92.62
MITM	98.05	88.69	92.37	90.50	89.43

Scanning	98.23	91.63	91.03	91.33	90.34
Ransomware	98.37	94.45	89.63	91.98	91.11
Password	98.33	92.15	91.24	91.69	90.77
XSS	98.50	93.19	91.97	92.57	91.74
Normal	98.50	91.23	93.98	92.59	91.76
Average	98.42	92.11	92.10	92.09	91.22

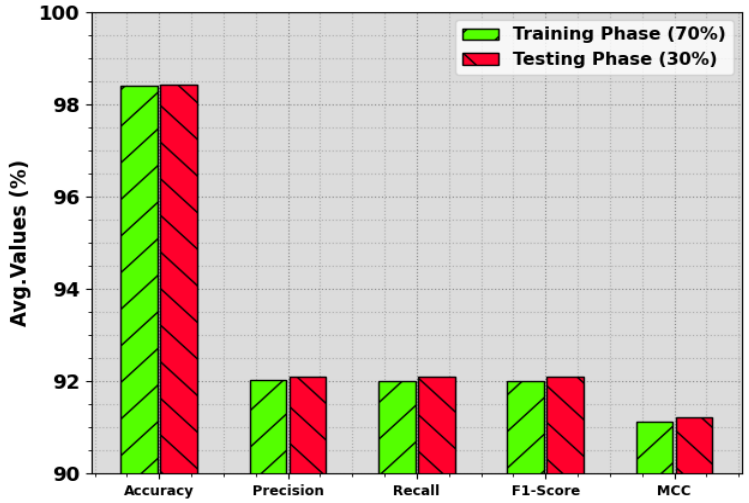

Figure 8. Average of RZDAD-ODL approach with 70:30 of TRAS/TESS

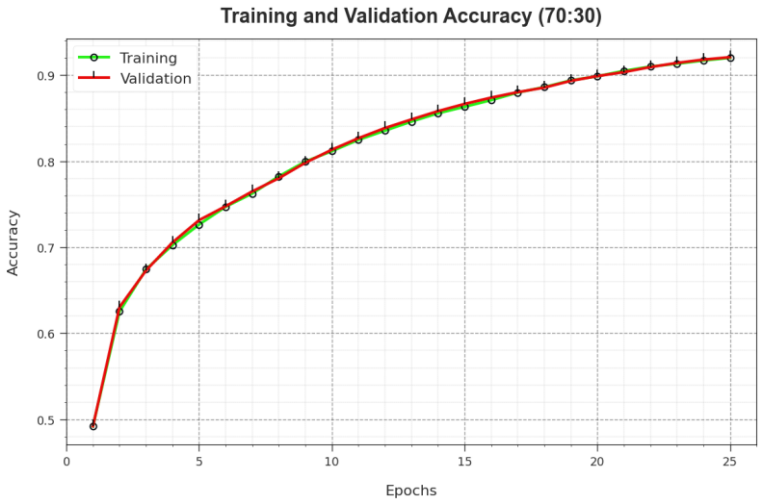

Figure 9. $Accu_y$ Curve of RZDAD-ODL approach with 70:30 of TRAS/TESS

As demonstrated in Fig. 9 we have produced accuracy curves for the TRAS and TESS to calculate the efficacy of the RZDAD-ODL technique with 70:30 of TRAS/TESS. This curve delivers valuable insight into the model's learning growth and its capacity to generalize. A notable progress in TRA and TES accuracy curves turn out to be apparent as we upsurge the epoch counts. This improvement designates the capability of the model to better detect models within the TRA and TES databases.

Fig. 10 shows a brief description of the RZDAD-ODL method with 70:30 of TRAS/TESS, the model's loss values during the TRA model. The declining trend in TRA loss over epochs designates that the model repeatedly improves its weights to minimalize prediction errors on TRA and TES datasets. This reveals how well the system fits the TRA data. Notably, the TRA and TES loss reliably drop, representing the model's effectual learning of patterns existing in both databases. Additionally, it depicts the model's alteration in diminishing alterations among forecasts and the original TRA labels.

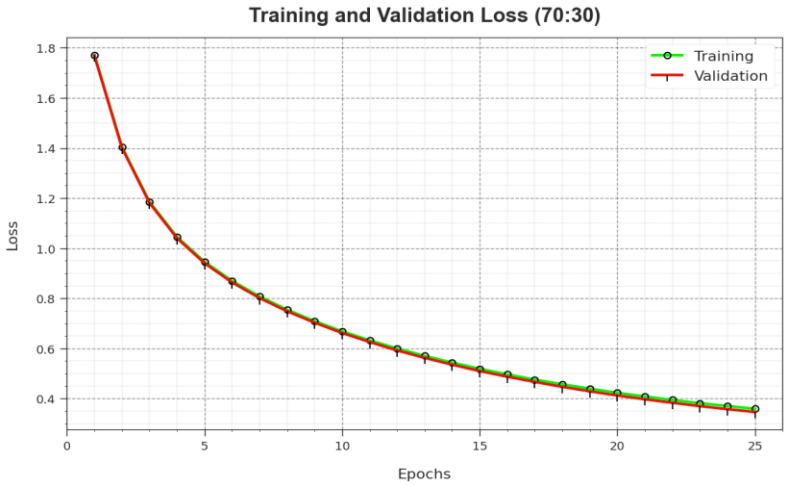

Figure 10. ROC curve of RZDAD-ODL approach with 70:30 of TRAS/TESS

In Table 4 and Fig. 11, a comprehensive study of the RZDAD-ODL approach with existing techniques takes place [23]. The outcomes illustrate that the CNN, DT, and AdaBoost models obtain poor performance over other models. Along with that, the LR, NB, XGBoost, and DenseNet models attain slightly boosted results. Although the NB model gains reasonable results over other ones, the RZDAD-ODL technique demonstrates promising results over other models with a maximum $accu_y$ of 98.48%, $prec_n$ of 92.43%, $reca_l$ of 92.41%, and F_{score} of 92.40%.

Table 4: Comparative outcome of RZDAD-ODL technique with other approaches

Methods	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$
LR Model	94.08	84.83	89.61	83.54
NB Algorithm	96.88	85.59	87.31	87.33
AdaBoost	90.67	86.01	88.22	85.48
CNN Motel	89.14	83.62	87.54	84.38
DT Model	89.93	91.11	84.08	84.28
XGBoost	95.70	85.79	89.60	82.85
DenseNet	94.75	85.77	83.40	85.92
RZDAD-ODL	98.48	92.43	92.41	92.40

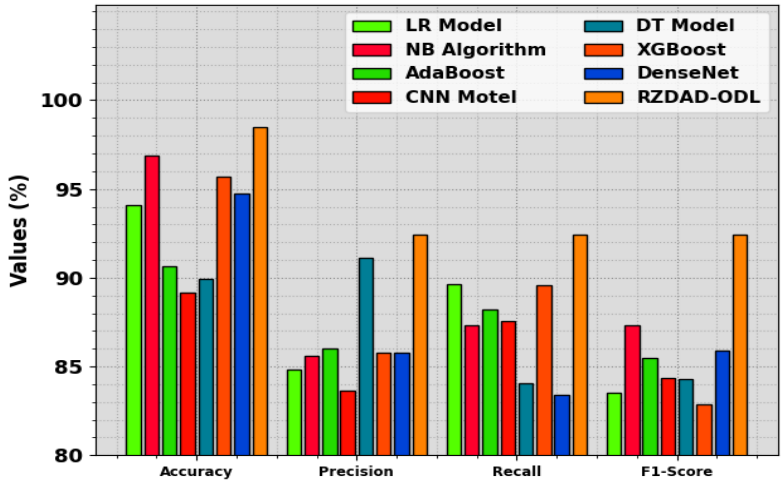

Figure 11. Comparative outcome of RZDAD-ODL technique with other approaches

Thus, the RZDAD-ODL model can be used for the automatic detection of zero-day attacks.

5. Conclusion

In this study, we have projected an RZDAD-ODL system for the IoT framework. The primary goal of the RZDAD-ODL technique lies in the automatic and effectual recognition of zero-day attacks in the IoT platform. In the presented RZDAD-ODL technique, HBA can be applied for the optimum range of the features. Besides, the RZDAD-ODL technique exploits the CVAE model for attack detection and its parameter tuning process can be performed by using ROA. The experimental study of the RZDAD-ODL model can be authorized on a benchmark dataset. Extensive comparison studies reported the better attack detection performance of the RZDAD-ODL method over other existing techniques.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. A. Tawalbeh, "Zero-day attack detection: a systematic literature review," *Artificial Intelligence Review*, pp. 1–79, 2023.

[2] U. Zahoora, M. Rajarajan, Z. Pan, and A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting-based ensemble classifier," *Applied Intelligence*, vol. 52, no. 12, pp. 13941–13960, 2022.

[3] V. Kumar and D. Sinha, "A robust intelligent zero-day cyber-attack detection technique," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2211–2234, 2021.

[4] S. I. Stellios, P. Kotzanikolaou, and M. Psarakis, "Advanced persistent threats and zero-day exploits in the industrial Internet of Things," in *Security and Privacy Trends in the Industrial Internet of Things*, pp. 47–68, 2019.

[5] M. Sarhan, S. Layeghy, M. Gallagher, and M. Portmann, "From zero-shot machine learning to zero-day attack detection," *International Journal of Information Security*, pp. 1–13, 2023.

[6] M. Swathy Akshaya and P. Ganapathi, "A review of machine learning methods applied for handling zero-day attacks in the cloud environment," 2020.

[7] B. M. Serinelli, A. Collen, and N. A. Nijdam, "On the analysis of open source datasets: validating IDS implementation for well-known and zero-day attack detection," *Procedia Computer Science*, vol. 191, pp. 192–199, 2021.

[8] A. Gorbenko and V. Popov, "Zero-Day attacks detection using an analysis of mobile robot motor primitives," in *2022 International Russian Automation Conference (RusAutoCon)*, pp. 278–283, IEEE, 2022.

[9] S. Ali, S. U. Rehman, A. Imran, G. Adeem, Z. Iqbal, and K. I. Kim, "Comparative evaluation of AI-based techniques for zero-day attacks detection," *Electronics*, vol. 11, no. 23, p. 3934, 2022.

[10] A. E. Topcu, Y. I. Alzoubi, E. Elbasi, and E. Camalan, "Social media zero-day attack detection using TensorFlow," *Electronics*, vol. 12, no. 17, p. 3554, 2023.

[11] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930–3944, 2021.

[12] S. J. Bu and S. B. Cho, "Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection," *Electronics*, vol. 10, no. 12, p. 1492, 2021.

[13] B. I. Hairab, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks," *IEEE Access*, vol. 10, pp. 98427–98440, 2022.

[14] A. K. Shukla, "An efficient hybrid evolutionary approach for identification of zero-day attacks on wired/wireless network systems," *Wireless Personal Communications*, vol. 123, no. 1, pp. 1–29, 2022.

[15] X. Cheng, J. Zhang, Y. Tu, and B. Chen, "Cyber situation perception for Internet of Things systems based on zero-day attack activities recognition within advanced persistent threat," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 16, p. e6001, 2022.

[16] S. Guo, T. Sivanthi, P. Sommer, M. Kabir-Querrec, N. Coppik, E. Mudgal, and A. Rossotti, "A zero-day container attack detection based on ensemble machine learning," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, IEEE, 2023.

[17] H. Hindy, R. Atkinson, C. Tachtatzis, J. N. Colin, E. Bayne, and X. Bellekens, "Towards an effective zero-day attack detection using outlier-based deep learning techniques," *arXiv Preprint*, 2020.

[18] W. Haider, N. Moustafa, M. Keshk, A. Fernandez, K. K. R. Choo, and A. Wahab, "FGMC-HADS: Fuzzy Gaussian mixture-based correntropy models for detecting zero-day attacks from Linux systems," *Computers & Security*, vol. 96, p. 101906, 2020.

[19] Y. Luo and Y. Hu, "The coverage improvement of the wireless sensor network based on the parameters optimized Honey Badger Algorithm," *IEEE Access*, 2023.

[20] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, "Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," *IEEE Access*, vol. 8, pp. 47072–47081, 2020.

[21] J. Grekow, "Generating polyphonic symbolic emotional music in the style of Bach using convolutional conditional variational autoencoder," *IEEE Access*, 2023.

[22] S. Baswaraju, V. U. Maheswari, K. K. Chennam, A. Thirumalraj, M. P. Kantipudi, and R. Aluvalu, "Future food production prediction using AROA-based hybrid deep learning model in agri-sector," *Human-Centric Intelligent Systems*, pp. 1–16, 2023.

[23] B. I. Hairab, H. K. Aslan, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection of zero-day attacks based on CNN and regularization techniques," *Electronics*, vol. 12, no. 3, p. 573, 2023.