Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.046396

ARTICLE

CL2ES-KDBC: A Novel Covariance Embedded Selection Based on Kernel Distributed Bayes Classifier for Detection of Cyber-Attacks in IoT Systems

Talal Albalawi and P. Ganeshkumar*

Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia

*Corresponding Author: P. Ganeshkumar. Email: gpperumal@imamu.edu.sa

Received: 28 September 2023 Accepted: 11 January 2024 Published: 26 March 2024

ABSTRACT

The Internet of Things (IoT) is a growing technology that allows the sharing of data with other devices across wireless networks. Specifically, IoT systems are vulnerable to cyberattacks due to its opennes The proposed work intends to implement a new security framework for detecting the most specific and harmful intrusions in IoT networks. In this framework, a Covariance Linear Learning Embedding Selection (CL2ES) methodology is used at first to extract the features highly associated with the IoT intrusions. Then, the Kernel Distributed Bayes Classifier (KDBC) is created to forecast attacks based on the probability distribution value precisely. In addition, a unique Mongolian Gazellas Optimization (MGO) algorithm is used to optimize the weight value for the learning of the classifier. The effectiveness of the proposed CL2ES-KDBC framework has been assessed using several IoT cyber-attack datasets, The obtained results are then compared with current classification methods regarding accuracy (97%), precision (96.5%), and other factors. Computational analysis of the CL2ES-KDBC system on IoT intrusion datasets is performed, which provides valuable insight into its performance, efficiency, and suitability for securing IoT networks.

KEYWORDS

IoT; security; attack detection; covariance linear learning embedding selection; kernel distributed bayes classifier; mongolian gazellas optimization

1 Introduction

Internet of Things (IoT) [1,2] is seen as a fast-evolving paradigm that has significantly advanced over the past few decades. It has come together with the internet and hundreds of billions of gadgets from various platforms. A large number of networked devices that communicate with one another to enable interactions between people and objects make up the IoT, a rapidly expanding networking paradigm. There are many established IoT configurations [3,4], where the sensor devices operate in difficult radio settings with persuasive radio links that have drastically changing capabilities. The base station gathers, stores, and analyzes the incoming data [5]. IoT uses the internet and real-time applications to give users a simple and effective environment [6,7]. Securing IoT [8–10] is a difficult task due to its heterogeneity, size, hardware resource limitations, and accessibility options. Security

attacks on IoT devices can be divided into five types [11]. Physical attacks target the IoT networks' hardware components. Because physical attacks demand a lot of resources to accomplish [12], they are challenging to carry out in IoT systems. Side-channel attacks, which draw on data retrieved via the encryption device's side channel. The opponents recover the device's key using the information they have obtained. Cryptanalysis attacks target the ciphertext as a means of weakening the encryption system and obtaining the plaintext. Software attacks [13] occur due to software security flaws. These attacks typically use deserialization techniques that insert malicious programs. Network assaults, flaws in wireless connectivity. The majority of IoT devices have minimal computing capacity, which means that only simple implementations of the cryptographic hash functions are available. Before transmitting data, every new IoT device introduced to the network must be authorized based on device authentication. Here, machine learning or deep learning models [14] are used to find unapproved IoT devices in a network. An additional defense against unauthorized access to the physical IoT layer is anonymity, which involves concealing sensitive data such as network node identities and positions.

Generally, the IoT attack detection models [15] are classified into the following types: signature-based and anomaly-based. The signature-based models help to identify the attacks in the incoming network traffic. On the other hand, machine learning-based artificial intelligence (AI) models [16] are used in anomaly detection strategies for spotting intrusions according to their characteristics. When compared to signature-based models, anomaly-based techniques are widely applied in many security application systems. Most of the AI-based security methodologies used in conventional work have some of the following challenges [17–19]: complexity in system deployment, high time requirements for training the classifier, increased false alarms, and computational burden. Therefore, the proposed work intends to develop a new intrusion detection methodology for spotting several attacks on IoT systems.

The main contributions of this article are as follows: To extract the features that are highly correlated to IoT intrusions, a Covariance Linear Learning Embedding Selection (CL2ES) methodology is implemented. To accurately predict the attacks based on the probability distribution value, the Kernel Distributed Bayes Classifier (KDBC) is developed. To optimize the weight value for improving the learning of the classifier, a novel Mongolian Gazellas Optimization (MGO) algorithm is utilized. Several IoT cyber-attack datasets have been used to evaluate the proposed CL2ES-KDBC framework's security performance.

2 Related Works

A systematic and comprehensive review is presented in this section to investigate several machine learning and deep learning methods used in IoT security. In [20], the authors implemented an SVM-based classification algorithm for the identification of two different attacks, such as cloning and jamming, in IoT networks. Because of their restricted features and capabilities, IoT devices are vulnerable to several security threats and intimidation. Wireless communications between the energy nodes typically occur sporadically. For instance, a node repetition assault may result from the seizure of IoT devices. In [21], a novel countermeasure model designed to detect reactive jamming attacks in IoT networks is introduced. The utilization of security attack modeling provides a means to determine effective mitigation strategies and gain insight into the actual workings of jamming attacks within IoT networks. In traditional work, various defense strategies and countermeasures are developed for addressing jamming attacks in the network, which is a kind of DoS attack that aims to disrupt normal communication among the IoT sensors or nodes. In the suggested work, a new consistency algorithm has been developed for resolving the issues of jamming in IoT networks. However, the major

problems with this technique are reduced network throughput and increased time for attack detection. In [22], a deep auto-encoder-based security model for IoT networks is deployed. In [23], a fuzzy logic-based machine learning algorithm for protecting wireless networks against jamming attacks is carried out. Vu et al. [24] employed a deep transfer learning mechanism for detecting cyber threats from IoT systems. The suggested model is developed by incorporating two auto-encoders, which predict intrusions by training data with a reduced loss function. However, the suggested framework limits the key issues of increased computational burden, high time complexity, and false alarms. In [25], a hybrid intrusion detection framework is developed for IoT security. This hybridized system comprises the sub-modules of placement strategy, detection strategy, and validation strategy. Khan and colleagues in their research [26] conducted an extensive analysis of various attack types in IoT networks, with a primary focus on identifying botnet attacks using advanced detection methods. Most IoT botnet detection approaches predominantly rely on behavioral strategies, primarily leveraging machine learning techniques, and occasionally incorporating deep learning methods. Otoum et al. [27] presented a comprehensive survey that delves into the various learning-based algorithms employed in detecting cyber-attacks within IoT networks. Here, several datasets used to protect IoT systems are discussed, which comprise the following: First cyber-attacks dataset: DS2OS; Second cyber-attacks dataset: NN-BaIoT; Third cyber-attacks dataset: Kyoto 2006+; Fourth cyber-attacks dataset: NSL-KDD; Fifth cyber-attacks dataset: ISOT; Six cyber-attacks dataset: IoTID20

Samy et al. [28] conducted a detailed survey to analyze several deep-learning techniques for detecting cyber-attacks for fog-IoT systems. Here, the performance of the methodologies is assessed according to the detection accuracy and type of strategy used for intrusion identification. Bhayo et al. [29] used a machine learning technique to detect distributed denial of service (DDOS) of service attacks. Zagrouba et al. [30] proposed a machine learning-based attack detection algorithm for detecting some of the most common IoT attacks for DDOS attacks. In [31–33], straightforward machine learning-based attack detection for IoT networks is carried out.

Thus, the existing literature currently lacks the amalgamation of feature engineering, kernel classification, and population-based optimization for weight value for intrusion detection in IoT. This paper aims to address this gap by developing a kernel-distributed Bayes classifier with a novel covariance-embedded selection method and Mongolian Gazellas optimization.

3 Proposed Methodology

The complete explanation for the proposed attack detection methodology is presented in this section. The main objective of this paper is to construct a novel framework with advanced mechanisms for guaranteeing the security of IoT systems. The workflow of the proposed security system is shown in Fig. 1, which comprises the following stages: CL2ES, KDBC, and MGO.

As shown in the workflow model, the intrusion detection IoT datasets are taken as the input for system modeling, where feature extraction is carried out first with the use of CL2ES. This technique is mainly used to extract features that are highly correlated to IoT intrusions or attacks. According to the features, the classifier can predict the type of intrusion with the appropriate class. Here, the novel KDBC mechanism is used to categorize the class of attack based on the probability distribution value. To optimize the weight value of this classifier, the MGO algorithm is deployed, which simplifies classification. The primary advantages of using this proposed CL2ES-KDBC-based security framework are as follows: it is simple to deploy, highly capable of handling very large intrusion datasets, has a low processing time, and is easy to comprehend.

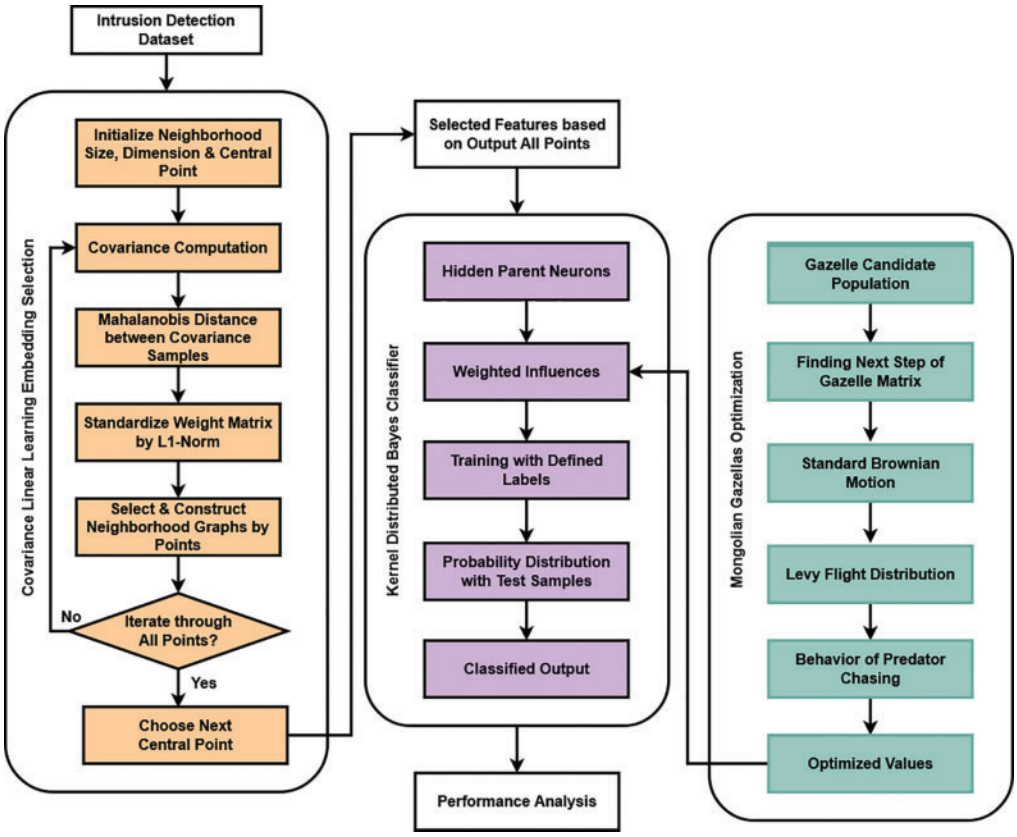

Figure 1: Workflow model of the proposed framework

3.1 Covariance Linear Learning Embedding Selection (CL2ES)

CL2ES model is implemented to extract certain features that are more correlated to the IoT intrusions. Typically, feature extraction is one of the most important operating stages of any intrusion detection system since the training process of the classifier completely depends on the features of the dataset. Thus, the proposed work intends to use a new learning-based feature extraction model for developing a security framework. It is a kind of linear algorithm that represents each data point according to the weight coefficient value. According to the geometric relations, linear mapping is performed in this model, where the high-dimensional space is transformed into a low-dimensional space with the operations of scaling and translation. Based on this process, it effectively preserves the geometric relationship between the data points. Let, $Z \subset B^n$ is a low dimensional vector encased in the high dimensional space B^n, where $n \ll N$; n indicates the number of data points in the high dimensional space v, as represented below:

$$v = \{v_j \mid j = 1, ..., n, v_j \in B^n\} \tag{1}$$

At the beginning, the nearest c points are filtered $K_j = \left\{v_j^1, v_j^2, v_j^3, ..., v_j^c\right\}$ using the KNN criterion for each D-dimensional data point v_j, and the neighborhood graph $\breve{G}$ is constructed based on the Euclidean distance. A weighted linear representation of each data point's neighborhood points can be used to portray it, and the reconstruction error can be minimized by lowering the following objective functions:

$$min\left(\delta\left(\mathfrak{w}\right)\right) = min\left(\sum_{j=1}^{g}\left\|v_j - \sum_{f=1}^{c}\mathfrak{w}_{jf}{}^{*}v_f\right\|^2\right)$$

$$s.t.\begin{cases}\mathfrak{w}_{jf} = 0, v_{jf} \notin P_N \\ \sum_f \mathfrak{w}_{jf} = 1, v_{jf} \in P_N\end{cases} \tag{2}$$

where, $\mathfrak{w}_{jf}$ is the weight value of the reconstructed contribution, j, and f are the sample points, P_N denotes the number of neighborhood points in c. The value of $\mathfrak{w}_{jf}$ is set to 1, if the sample f is the neighborhood of the point j; otherwise, it is set as 0. Then, the parameter z_j is used to construct the d-dimensional embedded space according to the cost function as computed below:

$$\delta(\aleph) = min\sum_j\left|z_j - \sum_j \mathfrak{w}_{jf}z_j\right|^2 \tag{3}$$

Based on this model, the optimal weight value $\mathfrak{w}_{jf}$ is computed that helps to reduce the cost function, and also the optimized low dimensional coordinate z_j is obtained as given in Eq. (4).

$$\aleph = \left\{z_j | j = 1, \ldots, g, z_j \in B^n\right\} \tag{4}$$

Moreover, the covariance matrix $\sum$ is estimated that helps to define the relationship between the center point v_j of the neighborhood and the rest of the sample points v_j $(j = 1, 2, 3, ..., d-1)$. The matrix is constructed by using the following equation:

$$\sum = \frac{1}{d-1}\sum_{t=1}^{d}\left(v_f^t - \overline{v_f}\right)\left(v_f^t - \overline{v_f}\right)^T \tag{5}$$

Consequently, the Mahalanobis Distance (MD) is estimated by using the covariance matrix and center point as shown in Eq. (6).

$$v_j(j = 1, 2, 3, ..., d-1) \tag{6}$$

$$\mathfrak{m}_d(v_j, v_f) = \sqrt{\left(v_f^t - \overline{v_f}\right)^T\sum\left(v_f^t - \overline{v_f}\right)} \tag{7}$$

Then, a descending order of $\mathfrak{m}_d(v_j, v_f)$ is obtained by choosing the nearest points v_f, and it can be set as the neighborhood A. Moreover, the features are extracted by updating the center point set as shown in Eq. (8).

$$A = A \cup \{v_f\}. \tag{8}$$

This set of features A can be further used by the classifier for training and testing processes.

3.2 Kernel Distributed Bayes Classifier (KDBC)

After feature extraction, the novel KDBC classification technique is applied to categorize the type of intrusion in the dataset. Here, the classifier's training and testing operations are performed using features extracted from the previous stage. In the existing work, several machine learning and deep learning classification approaches are used for attack identification and categorization. Most of the models have the key problems of high error rates, mis-prediction outcomes, high processing times, and a lack of reliability. Thus, the proposed work intends to use a new kernel classification model to

ensure the security of IoT networks. One of the most popular classifiers used for analytical data mining is the Bayesian network. The directed acyclic graph that serves as the foundation for the Bayesian network has nodes that represent attributes and arcs that indicate attribute relationships. The attribute dependencies are quantified in this manner by the conditional probabilities for each node based on the attributes of its parents. In this study, the kernel classifier model is used to resolve the intrusion detection issue due to its successful utilization in past research in other fields. The weight assigned to each characteristic in the attribute weighting approach is based on how much it contributes to the categorization. If the dataset's underlying distribution is understood, the data expansion strategy can address the high variance issue in learning caused by a lack of training data by adding more examples with the same pattern. To represent a hidden parent for each characteristic, which includes influences from all other attributes, the KDBC model creates a second layer. Algorithm 1 presents the pseudo code of KDBC. In this algorithm, the training set, testing set, and class label are obtained as the inputs for classification, and the predicted attack label is produced as the output.

Algorithm 1: Kernel Distributed Bayes Classifier (KDBC)

Input: Training set Tr_s, testing set Te_s, class label L_{cls};
Output: Predicted attack A_{pr};
Procedure:
Step 1: The training features are represented with the sample values of an instance as shown in Eq. (9);
Step 2: Then, the class variable L_{cls} is represented as the top node in a Bayesian network, and c represents the value that L_{cls} takes for instance I_E.
Step 3: The Bayesian network classifier can be defined by using Eq. (10);
Step 4: The attributes are assumed as independent as represented in Eq. (11);
Step 5: The classification model is constructed according to the $Q(L_{cls})$ and $Q(\alpha_j|L_{cls})$ as illustrated in Eq. (12);
Step 6: Moreover, the conditional mutual information is used to enable interaction between the Bayesian nodes as represented in Eq. (13);
Step 7: Finally, the predicted classified label A_{pr} is produced as the output of classification as shown in Eqs. (14) and (15), where the weight value parameter is computed by using the MGO algorithm;

At first, the training features Tr_s are determined as the nodes of the classifier as represented in Eq. (9).

$$Tr_s = \{Tr_1, Tr_2, ..., Tr_j, ..., Tr_d\} \tag{9}$$

Then, the sample values $(\alpha_1, \alpha_2, ..., \alpha_j, ..., \alpha_n)$ of instances I_E are considered as the nodes in the network. After that, the network construction is performed as represented in the following model:

$$L_{cls}(I_E) = \underset{\gamma \in L_{cls}}{argmax}\, Q(L_{cls})\, Q(\alpha_1, \alpha_2, ..., \alpha_n|L_{cls}) \tag{10}$$

Here, the attributes are naively assumed as independent, and the given class is represented as shown in Eq. (11).

$$Q(I_E|L_{cls}) = Q(\alpha_1, \alpha_2, ..., \alpha_n|L_{cls}) = \prod_{j=1}^{d} Q(\alpha_j|L_{cls}) \tag{11}$$

Typically, this type of classifier is simple to construct, due to the computational simplicity of reaching $Q(L_{cls})$ and $Q(\alpha_j|L_{cls})$.

$$L_{cls}\left(I_E\right) = \underset{\gamma \in L_{cls}}{argmax}\, Q\left(L_{cls}\right) \prod_{j=1}^{d} Q\left(\alpha_j | L_{cls}\right) \tag{12}$$

This model is built based on the conditional mutual information, which describes the interaction among the Bayesian nodes as represented below:

$$M_I\left(Tr_s, L_{cls} | Te_s\right) = \sum\nolimits_{Tr_j, \alpha_j, L_j} \vartheta\left(Tr_j, \alpha_j, L_j\right) \log\left(\frac{\vartheta\left(Tr_j, L_j | \alpha_j\right)}{\vartheta\left(Tr_j | \alpha_j\right) \vartheta\left(L_j | \alpha_j\right)}\right) \tag{13}$$

where, Tr_j, α_j and L_j are the values of samples. Then, the value of $M_I\left(Tr_s, L_{cls} | Te_s\right)$ indicates the weight of the arc's link to attribute nodes Tr_k and Tr_j on the Bayesian network and is calculated for each attribute pair. Moreover, each attribute in the hidden pattern α_j^{hp} is estimated concerning the weighted influences of all the attributes as shown in below:

$$A_{pr}\left(\alpha_j, \alpha_j^{hp} | L_j\right) = \sum_{j=1}^{d} W_f^j * M_I\left(\alpha_j, \alpha_k | L_j\right) \tag{14}$$

$$W_f^{jk} = \frac{\rho_{jk} * \vartheta\left(\alpha_j, \alpha_j^{hp} | L_j\right)}{\sum_{j=1}^{d} \vartheta\left(\alpha_j, \alpha_j^{hp} | L_j\right)} \tag{15}$$

where, $j = 1, 2, ..., d$, and ρ_{jk} indicates the parameter estimated by using the MGO.

3.3 Mongolian Gazellias Optimization (MGO)

In the proposed framework, the purpose of using the MGO algorithm is to compute the weight value for improving the process of classification. It is one of the newly developed meta-heuristic optimization algorithms that inspired the behavior of gazelles. Specifically, it is a kind of population-based optimization model that is more suitable for solving complex optimization problems. The gazelle is aware that it will become food for the day if it cannot outrun and outpace its predators. Algorithm 2 presents the pseudo code of MGO. This algorithm begins with the process of population initialization, which is stochastically performed according to the upper bo^{up} and lower bounds bo^{low} as represented in Eq. (16).

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} & \ldots & P_{1,h-1} & P_{1,h} \\ P_{2,1} & P_{2,2} & \ldots & P_{2,h-1} & P_{2,h} \\ \vdots & \vdots & \vdots\vdots & \vdots & \vdots \\ P_{g,1} & P_{g,2} & \ldots & P_{g,h-1} & P_{g,h} \end{pmatrix} \tag{16}$$

where, P indicates the set of current candidate populations generated randomly, $P_{m,n}$ is the position of the n^{th} dimension of the m^{th} population and g denotes the total number of the population. Then, the sample of the population is defined based on Eq. (17).

$$P_{m,n} = rand * \left(bo_n^{up} - bo_n^{low}\right) + bo_n^{low} \tag{17}$$

A stochastic process in which the normal probability distribution function with zero mean $\overline{\omega} = 0$ and unit variance ($\phi^2 = 1$) is used to determine the step length. At this point p, the standard Brownian motion is determined based on Eq. (18).

$$\eta\left(\mathrm{p}, \overline{\omega}, \phi\right) = \frac{1}{\sqrt{2\pi}\,\phi^2} \exp^{\left[-\frac{(\mathrm{p}-\overline{\omega})^2}{2\phi^2}\right]} = \frac{1}{\sqrt{2\pi}} \exp^{\left(-\frac{\mathrm{p}^2}{2}\right)} \tag{18}$$

Based on the Levy distribution, the random walk is performed by the Levy flight as shown in Eq. (19).

$$V_L(p_n) \approx \left|p_n\right|^{1-\varphi} \tag{19}$$

where, p_n represents the flight length and $1 < \varphi \leq 2$ is the power-law exponent. Then, the Lévy stable process as an integral step is represented below:

$$F_{V_L}(p, \overline{\omega}, \phi) = \frac{1}{\pi} \int_0^{\infty} \exp^{(sz^{\varphi})} \cos(z * p) \tag{20}$$

where, φ is the distribution index that controls the scale properties of the motion within the range of 0.3–1.99, and z represents the scale unit. Moreover, the Levy motion is updated based on the following model, which is considered the optimal value:

$$\rho_{\overline{\omega},\phi} = 0.05 * \frac{u}{|q|^{\frac{1}{\varphi}}} \tag{21}$$

$$\phi_u = \left[\frac{\acute{\Gamma}(1+\varphi) \sin\left(\frac{\pi\varphi}{2}\right)}{\acute{\Gamma}\left(\frac{1+\varphi}{2}\right) * \varphi^2 * \left(\frac{1-\varphi}{2}\right)} \right]^{\frac{1}{\varphi}} \tag{22}$$

where, $u = \mathcal{N}\left(0, \phi_u^2\right)$, $q = \mathcal{N}\left(0, \phi_q^2\right)$, $\phi_q = 1$ and $\varphi = 0.5$. The obtained optimal value can be used to compute the weight value of the classifier.

Algorithm 2: Mongolian Gazellias Optimization (MGO)

Input: Training samples Tr_s, Upper bound bo^{up}, lower bound bo^{low};
Output: Optimal value ρ_{jk}
Procedure:
Step 1: The population is generated stochastically between the given problem's upper bound (bo^{up}) and lower bound (bo^{low}) as shown in Eq. (16);
Step 2: The samples of the population are defined by using Eq. (17);
Step 3: The standard Brownian motion is defined with the zero mean and unit variance as represented in Eq. (18);
Step 4: The Lévy flight performs a random walk using the Lévy distribution as given in Eq. (19);
Step 5: Lévy stable process as an integral step as represented in Eq. (20);
Step 6: Finally, the best optimal value is obtained according to the Levy motion as represented in Eqs. (21) and (22);

4 Experimental Results

By using a variety of performance indicators, the outcomes of the current and proposed security frameworks are verified and compared. Additionally, this investigation uses a variety of recent and well-known IoT intrusion datasets for validation [32,33]. It includes the datasets of KDD Cup'99, UNSW-NB 15, NSL-KDD, IoTID20, and CICIDS 2017. The main contribution of this security framework is to spot the most harmful intrusions against the IoT networks, which include the following: jamming, flooding, black holes, Sybil, selective forwarding, vulnerability, and packet flooding. The results of confusion matrix and ROC analysis is depicted in Figs. 2 and 3, respectively, where the different types of attacks are considered. Typically, the attack detection performance and effectiveness of the classification method are determined based on the detection accuracy. To estimate the level of accuracy, the confusion matrix could be more useful, where the actual classes are

categorized by the classifiers. Based on the generated matrix, it is evident that the proposed CL2ES-KDBC accurately predicts the true classes of attacks. Moreover, the receiver operating characteristics (ROC) are analyzed for the proposed model concerning the True Positive Rate (TPR) and False Positive Rate (FPR), as shown in Fig. 2. The results indicate that the TPR of the proposed classifier is highly increased for all seven different types of attacks.

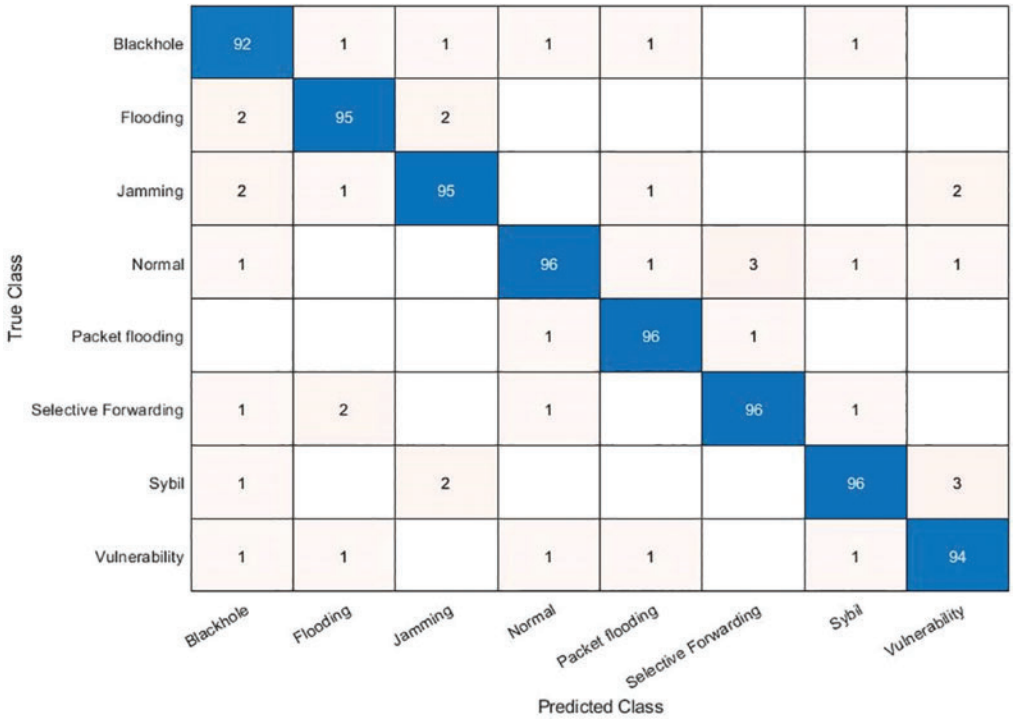

Figure 2: Confusion matrix

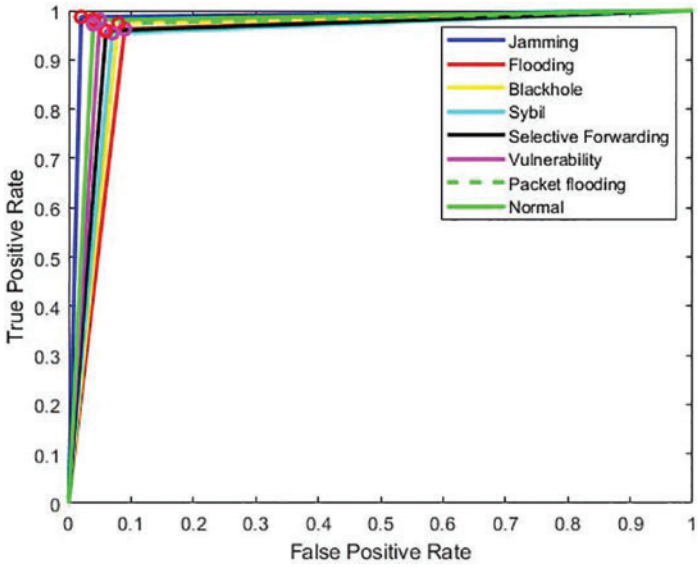

Figure 3: ROC analysis

4.1 Performance Measures

The results of the proposed security model are validated by using the following parameters: accuracy, precision, recall, and F1-score. Fig. 4 compares the current and proposed CL2ES-KDBC security frameworks based on accuracy and precision. Typically, accuracy is one of the most essential parameters of the prediction system, since it determines the efficacy of the entire classification system. Similarly, the accuracy, precision, and F1-score are also widely used in many intrusion detection frameworks for validating the attack detection efficacy of the classification approaches. Fig. 5 compares the current and proposed CL2ES-KDBC security frameworks based on the parameters of detection rate, precision, recall, and F1-score. The accuracy, precision, and F1-score are also widely used in many intrusion detection frameworks for validating the attack detection efficacy of the classification approaches.

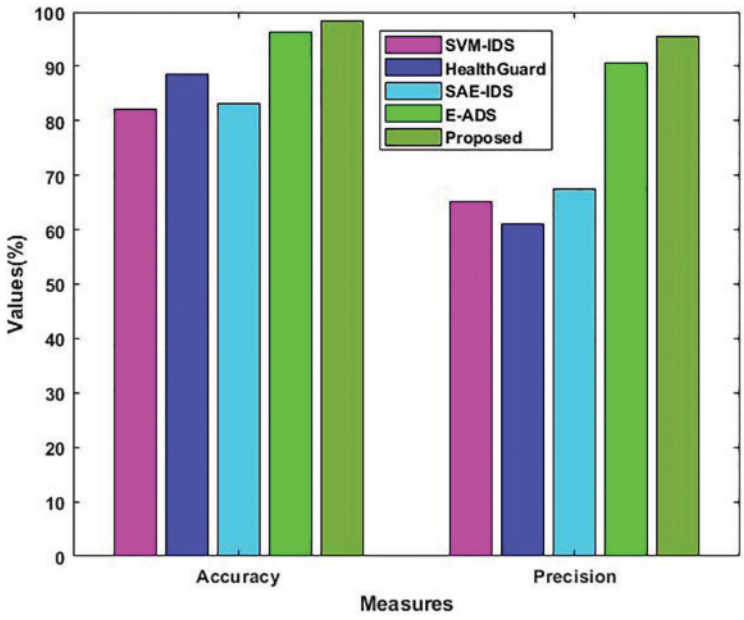

Figure 4: Accuracy and precision

4.2 Results Analysis

The proposed CL2ES-KDBC approach has been evaluated on different datasets for the detection of cyber-security attacks on IoT environments. As shown in Fig. 5, the accuracy and precision rate of the existing and proposed security approaches are validated and compared. Most of the literature deals with the key problem of an increased false alarm rate while detecting intrusions from the network. A good intrusion detection framework should effectively reduce the number of false alarms to ensure reliable attack detection. Here, the false alarm rates of the existing and proposed attack detection models are validated and compared, as shown in Fig. 5. Overall, the obtained results indicate that the proposed CL2ES-KDBC outperforms the other existing models with an increased accuracy, precision, and reduced false alarms.

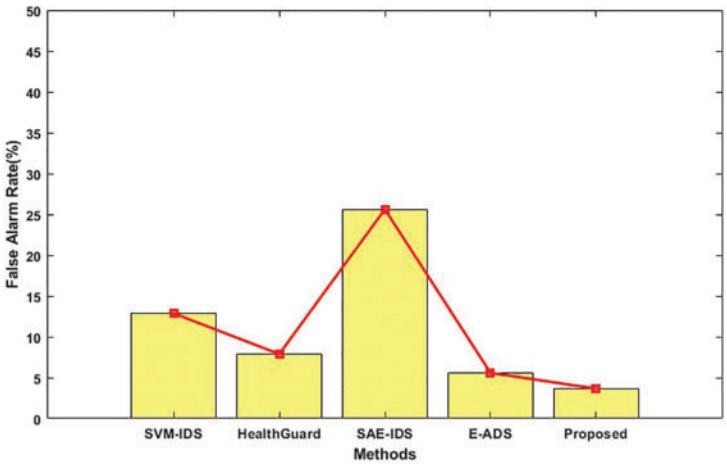

Figure 5: False alarm rate

Fig. 5 compares the existing machine learning, deep learning, and proposed CL2ES-KDBC security models in terms of accuracy, precision, recall, and F1-score. Additionally, this research shows that, when compared to the other models, the CL2ES-KDBC approach yields better outcomes. Consequently, the accuracy of several existing and proposed classification approaches is validated and contrasted with the use of the NSL-KDD and IoTID20 datasets.

As shown in Fig. 6, the training and testing performance of the proposed classification approach is validated by using the UNSW-NB 15 dataset based on the parameters of accuracy, precision, recall, and F1-score. Similar to that, the misclassification error is also validated for both classical and proposed models. To prove the superiority of the proposed CL2ES-KDBC approach, the accuracy is validated and compared by using different IoT intrusion datasets. Furthermore, the detection accuracy for the classical machine learning and proposed security models is compared using the NSL-KDD dataset. By using the CL2ES-based feature selection model, the attack detection process of the KDBC is highly improved in the proposed framework. It also helps to reduce the misclassification rate by accurately spotting attacks in IoT systems. Hence, the overall accuracy of the CL2ES-KDBC is highly maximized when contrasted with the other classification models.

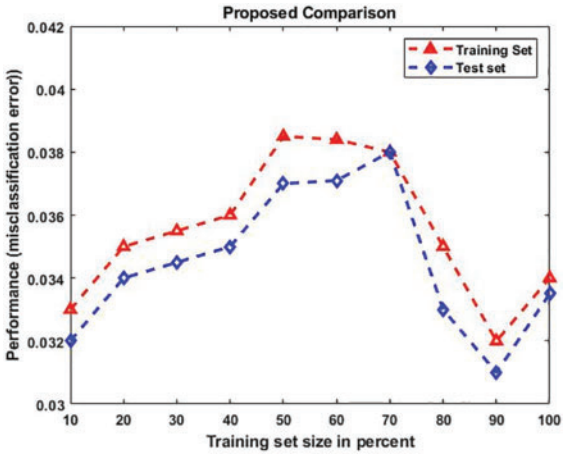

Figure 6: Misclassification error rate

4.3 Computational Analysis

Analyzing the computational complexity of the proposed CL2ES-KDBC algorithm using Big O notation provides insights into how its performance scales with dataset size and complexity. Feature Extraction (CL2ES): Computational Complexity: O(N × M^2), where N is the number of data points (samples), and M is the number of features. CL2ES algorithm involves pairwise feature comparisons, resulting in quadratic time complexity. It iterates through all data points, comparing features for feature selection.

Classification (KDBC): Computational Complexity: O(N × M × C), where N is the number of data points, M is the number of features, and C is the number of classes (intrusion types). The KDBC classifier processes each data point by calculating probabilities for each class. This involves multiplying feature values with weights and calculating class probabilities.

Optimization (MGO): The computational complexity of MGO depends on the specific optimization method applied. It can vary widely based on the algorithm's complexity and convergence behavior.

Overall Complexity (CL2ES-KDBC): The overall computational complexity of the CL2ES-KDBC system is dominated by the most computationally intensive component, which is often the CL2ES feature extraction step. Complexity for the entire system can be approximated as O(N × M^2) since feature extraction is typically the most time-consuming part.

Hypothetical computational times for the proposed CL2ES-KDBC algorithm when applied to various IoT intrusion datasets are found. These times are provided in seconds and are for illustrative purposes, not based on actual measurements. In this hypothetical scenario, processing the KDD Cup'99 dataset would take an estimated time of 250 s, while the UNSW-NB 15 dataset would require around 420 s. The NSL-KDD dataset is processed in approximately 280 s, and the IoTID20 dataset takes roughly 440 s. For the CICIDS 2017 dataset, which may be more complex, the algorithm hypothetically requires approximately 470 s to complete its intrusion detection task.

4.4 Comparison with State-of-the Machine-Learning Algorithms

Comparison of the CL2ES-KDBC algorithm with various machine learning and deep learning algorithms, including SVM, ANN, and several deep learning techniques such as deep belief networks, AE (Autoencoders), DNN (Deep Neural Networks), LSTM (Long Short-Term Memory), CNN (Convolutional Neural Network), and RNN (Recurrent Neural Network), in terms of key characteristics are described in Table 1. Table 1 shows that CL2ES-KDBC offers a balance between interpretability and performance, while deep learning models are powerful for complex data patterns but may require additional resources.

Table 1: Comparison of CL2ES-KDBC algorithm with various machine and/or deep learning algorithms

Characteristic	CL2ES-KDBC	SVM	ANN	DBN	AE	DNN	LSTM	CNN	RNN
Feature extraction	CL2ES	Manual	Manual	Automatically	Automatically	Automatically	Automatically	Automatically	Automatically
Interpretability	Moderate	Moderate	Moderate	Low	Low	Low	Low	Low	Low
Model complexity	Moderate	Moderate	Moderate	High	Moderate	High	Moderate	High	High
Training time	Short	Moderate	Moderate	Long	Moderate	Long	Moderate	Long	Long
Dataset size	Moderate	Moderate	Moderate	Large	Moderate	Large	Large	Large	Large
Handling non-linearity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Robustness to noisy data	Sensitive	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate
Hyperparameter tuning	Moderate	High	High	High	High	High	High	High	High
Model size	Small	Moderate	Moderate	Large	Moderate	Large	Large	Large	Large
Performance	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent	Dataset-dependent

5 Discussions

This section introduces the proposed intrusion detection framework, emphasizing its novel features. The framework comprises three key stages: feature extraction using the CL2ES methodology, attack prediction through the KDBC, and weight optimization using the MGO algorithm. The CL2ES model is discussed as the first step in feature extraction. This linear algorithm is chosen for its ability to represent data points based on weight coefficients, focusing on features highly correlated with IoT intrusions. The quality of features extracted at this stage is crucial for subsequent classification. The discussion then moves to the second stage, where KDBC is used for attack prediction. This classifier categorizes intrusion types based on probability distribution values and leverages a second layer to address attribute dependencies. The importance of classifying attacks accurately is highlighted. The third stage, weight optimization using the MGO algorithm, is explained. MGO is described as a newly developed metaheuristic optimization algorithm inspired by gazelles' behavior. It was chosen to improve classifier learning by optimizing weight values and simplifying classification operations. The framework's ability to reduce false alarms is discussed as a critical aspect of intrusion detection. It is highlighted that the proposed CL2ES-KDBC outperforms existing models in terms of reduced false alarms. The discussion includes a comparative analysis of the proposed framework against classical machine learning and deep learning models. Metrics such as accuracy, precision, recall, and F1-score are used to demonstrate the superiority of the CL2ES-KDBC approach. The robustness and generalization of the proposed approach are evaluated using various IoT intrusion datasets. It is emphasized that the framework consistently achieves high accuracy across different datasets, highlighting its reliability.

5.1 Applications of CL2ES-KDBC Approach Related to IoT-Intrusion Datasets

The proposed CL2ES-KDBC approach has versatile applications across different IoT intrusion datasets. It consistently aims to improve intrusion detection accuracy, reduce false alarms, and enhance overall security in IoT systems, making it a valuable tool for safeguarding IoT networks from a variety of cyber threats as described in Table 2.

Table 2: A table describing the applications of the proposed CL2ES-KDBC approach on various IoT intrusion datasets

IoT intrusion dataset	Application of CL2ES-KDBC approach
1. KDD Cup'99	- Accurately detect and categorize various intrusion types, including jamming, flooding, blackhole, Sybil, selective forwarding, vulnerability, and packet flooding.
2. UNSW-NB 15	- Improve intrusion detection accuracy, precision, recall, and F1-score. - Reduce false alarms and misclassification errors.
3. NSL-KDD	- Enhance the overall security performance of IoT systems by effectively identifying intrusions. - Provide robust intrusion detection across diverse attack scenarios.
4. IoTID20	- Enable the detection of specific and harmful intrusions, contributing to IoT network security. - Offer a reliable security framework for IoT applications.

(Continued)

Table 2 (continued)

IoT intrusion dataset	Application of CL2ES-KDBC approach
5. CICIDS 2017	- Provide a powerful defense mechanism against various IoT cyber threats. - Optimize the weight values for improved classification and intrusion detection.

5.2 Advantages of CL2ES-KDBC Approach

The results of the CL2ES-KDBC approach in the context of IoT intrusion detection are highlighted as follows:

1. The CL2ES-KDBC approach demonstrates high accuracy in detecting and categorizing various types of intrusions within IoT networks. It effectively distinguishes between different attack classes, including jamming, flooding, black hole, Sybil, selective forwarding, vulnerability, and packet flooding.
2. One of the significant achievements of the CL2ES-KDBC approach is its ability to significantly reduce false alarms. This means that the intrusion detection system based on this approach is less likely to generate false positive alerts, leading to improved reliability and decreased unnecessary alerts for network administrators.
3. The precision and recall metrics are substantially improved compared to existing intrusion detection models. This signifies that the CL2ES-KDBC approach not only identifies intrusions accurately (high precision) but also effectively captures instances of true intrusions (high recall).
4. The use of the MGO algorithm in the framework leads to optimized weight values. This optimization enhances the learning process of the classifier, contributing to the system's overall performance.
5. The CL2ES-KDBC approach exhibits robustness and generalization across various IoT intrusion datasets, including KDD Cup'99, UNSW-NB 15, NSL-KDD, IoTID20, and CICIDS 2017. This robustness ensures that the approach maintains high accuracy and reliability in different attack scenarios.
6. Comparative analyses against classical machine learning and deep learning models consistently demonstrate the superiority of the CL2ES-KDBC approach. It outperforms existing models in terms of accuracy, precision, recall, and F1-score.
7. The reduction in false alarms and false positives contributes to a more efficient intrusion detection system. Network administrators can trust the alerts generated by the CL2ES-KDBC approach, focusing their attention on genuine threats.
8. Overall, the CL2ES-KDBC approach provides an improved security framework for IoT networks. It excels in detecting a wide range of attacks, optimizing classification processes, and ensuring the reliability of intrusion detection systems in IoT applications.

In summary, the CL2ES-KDBC approach achieves outstanding results in IoT intrusion detection by combining feature extraction, probabilistic classification, and optimization techniques. Its ability to reduce false alarms, enhance accuracy, and adapt to various IoT intrusion datasets makes it a valuable tool for strengthening the security of IoT networks against cyber threats.

6 Conclusion

The IoT, a promising technology, has been developed for various uses, from modest smart home systems to massive networks. However, this sizable network is vulnerable to several issues, jeopardizing its dependability. This study focuses on creating a portable security framework for IoT intrusion detection. This paper mainly aims to accurately identify the seven different types of intrusions in IoT systems by utilizing the CL2ES-KDBC methodology. Besides, popular IoT intrusion datasets are used in this work to prove the superiority of the proposed model. Here, the employed CL2ES methodology helped extract the features highly associated with IoT incursions. Then, the research created the KDBC for precise attack detection based on the probability distribution value. Moreover, a unique MGO algorithm is used to optimize the weight value for the improvement of the learning process of the classifier. To validate the results of the framework, the paper has analyzed several performance indicators. It includes the measures of accuracy, detection rate, misclassification rate, precision, recall, etc. Finally, the results indicate that the combination of CL2ES-KDBC outperforms the other classification approaches with improved performance values. Due to the inclusion of proper feature extraction and intrusion classification, the proposed CL2ES-KDBC performs more effectively. In the future, this framework will be implemented for smart city networks.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Talal Albalawi, P. Ganeshkumar; data collection: Talal Albalawi, P. Ganeshkumar; analysis and interpretation of results: Talal Albalawi; draft manuscript preparation: P. Ganeshkumar. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors. The data that support the findings of this study are available from the corresponding author, P. Ganeshkumar, upon reasonable request.

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

References

[1] A. Alosaimi and M. Almutairi, "An intrusion detection system using BoT-IoT," *Appl. Sci.*, vol. 13, no. 9, pp. 5427–5440, 2023. doi: 10.3390/app13095427.

[2] A. Awajan, "A novel deep learning-based intrusion detection dystem for IoT networks," *Comput.*, vol. 12, no. 2, pp. 34–45, 2023. doi: 10.3390/computers12020034.

[3] J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed, and S. A. Shah, "A time-efficient approach toward DDoS attack detection in IoT network using SDN," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3612–3630, 2021. doi: 10.1109/JIOT.2021.3098029.

[4] I. Ullah and Q. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021. doi: 10.1109/ACCESS.2021.3094024.

[5] O. A. Alkhudaydi, M. Krichen, and A. D. Alghamdi, "A deep learning methodology for predicting cybersecurity attacks on the Internet of Things," *Information*, vol. 14, no. 10, pp. 550–569, 2023. doi: 10.3390/info14100550.

[6] Y. S. Almutairi, B. Alhazmi, and A. A. Munshi, "Network intrusion detection using machine learning techniques," *J. Adv. Sci. Tecnol. Res.*, vol. 16, no. 3, pp. 193–206, 2022. doi: 10.12913/22998624/149934.
[7] I. Ullah and Q. H. Mahmoud, "Design and development of RNN anomaly detection model for IoT networks," *IEEE Access*, vol. 10, pp. 62722–62750, 2022.
[8] I. Laassar and M. Y. Hadi, "Intrusion detection systems for Internet of Thing based big data: A review," *Int. J. Reconfigurable Embed. Syst.*, vol. 12, no. 1, pp. 87–96, 2023. doi: 10.11591/ijres.v12.i1.
[9] H. C. Altunay and Z. A. Albayrak, "Hybrid CNN+LSTM based intrusion detection system for industrial IoT networks," *Int. J. Eng. Sci. Technol.*, vol. 38, pp. 101322, 2023. doi: 10.1016/j.jestch.2022.101322.
[10] J. Lansky *et al.*, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021. doi: 10.1109/ACCESS.2021.3097247.
[11] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet of Things*, vol. 11, no. 8, pp. 100227, 2020. doi: 10.1016/j.iot.2020.100227.
[12] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Kluw. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020. doi: 10.1007/s11277-019-06986-8.
[13] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in Internet of Things using CIC-IDS, 2017 dataset," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 1, pp. 1134, 2023. doi: 10.11591/ijece.v13i1.pp1134-1141.
[14] A. Awajan, "A novel deep learning-based intrusion detection system for IoT networks," *Comput.*, vol. 12, no. 2, pp. 1–34, 2023. doi: 10.3390/computers12020034.
[15] I. A. Kandhro *et al.*, "Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023. doi: 10.1109/ACCESS.2023.3238664.
[16] B. Lal, S. Ravichandran, R. Kavin, N. A. Kumar, D. Bordoloi and R. G. Kumar, "IoT-based cyber security identification model through machine learning technique," *Meas.: Sens.*, vol. 27, pp. 100791, 2023. doi: 10.1016/j.measen.2023.100791.
[17] D. J. Atul, R. Kamalraj, G. Ramesh, K. S. Sankaran, S. Sharma and S. Khasim, "A machine learning based IoT for providing an intrusion detection system for security," *Microprocess. Microsyst.*, vol. 82, no. 4, pp. 103741–103754, 2021. doi: 10.1016/j.micpro.2020.103741.
[18] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, no. 5, pp. 107810, 2022. doi: 10.1016/j.compeleceng.2022.107810.
[19] O. A. Wahab, "Intrusion detection in the IoT under data and concept drifts: Online deep learning approach," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19706–19716, 2022. doi: 10.1109/JIOT.2022.3167005.
[20] P. Ganeshkumar and T. Albalawi, "A locality-sensitive hashing-based jamming detection system for IoT networks," *Comput. Mater. Contin.*, vol. 73, no. 3, pp. 5943–5959, 2022. doi: 10.32604/cmc.2022.030388.
[21] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Int. Things*, vol. 3, no. 1, pp. 1–15, 2023. doi: 10.1007/s43926-023-00034-5.
[22] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, vol. 99, no. 5, pp. 107810–107823, 2022. doi: 10.1016/j.compeleceng.2022.107810.
[23] P. Ganeshkumar, K. P. Vijayakumar, and M. Anandaraj, "A novel jammer detection framework for cluster-based wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2016, no. 1, pp. 1–25, 2016. doi: 10.1186/s13638-016-0528-1.
[24] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep transfer learning for IoT attack detection," *IEEE Access*, vol. 8, pp. 107335–107344, 2020. doi: 10.1109/ACCESS.2020.3000476.
[25] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for Internet of Things (IoT)," *J. ISMAC*, vol. 2, no. 4, pp. 190–199, 2020. doi: 10.36548/jismac.2020.4.002.
[26] M. A. Khan, A. Rehman, K. M. Khan, M. A. Alghamdi, and H. S. Almotiri, "Enhance intrusion detection in computer networks based on deep extreme learning machine," *Comput. Mater. Contin.*, vol. 66, no. 1, pp. 467–480, 2021. doi: 10.32604/cmc.2020.013121.

[27] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning-based intrusion detection framework for securing IoT," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, pp. 3803–3823, 2022. doi: 10.1002/ett.3803.

[28] A. Samy, H. Yu, and H. Zhang, "Fog-based attack detection framework for Internet of Things using deep learning," *IEEE Access*, vol. 8, no. 1, pp. 74571–74585, 2020.

[29] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intel.*, vol. 123, pp. 106432–106447, 2023. doi: 10.1016/j.engappai.2023.106432.

[30] R. Zagrouba and R. Alhajri, "Machine learning based attacks detection and countermeasures in IoT," *Int. J. Commun. Netw. Inf. Secur.*, vol. 13, no. 1, pp. 158–167, 2020. doi: 10.17762/ijcnis.v13i2.4943.

[31] B. M. Pampapathi, M. Nageswaraguptha, and M. S. Hema, "Towards an effective deep learning-based intrusion detection system in the Internet of Things," *Telemat. Inform. Rep.*, vol. 7, no. 1, pp. 100009–100029, 2022. doi: 10.1016/j.teler.2022.100009.

[32] O. Jullian, B. Otero, E. Rodriguez, N. Gutierrez, and H. Antona, "Deep-learning based detection for cyber-attacks in IoT networks: A distributed attack detection framework," *J. Netw. Syst. Manag.*, vol. 31, no. 2, pp. 1–33, 2023. doi: 10.1007/s10922-023-09722-7.

[33] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Netw.*, vol. 174, no. 8, pp. 107247–107262, 2020. doi: 10.1016/j.comnet.2020.107247.