Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4989410, 12 pages
https://doi.org/10.1155/2021/4989410

Research Article

A Novel, Efficient, and Secure Anomaly Detection Technique Using DWU-ODBN for IoT-Enabled Multimedia Communication Systems

M. Sathya,[1] M. Jeyaselvi,[2] Lalitha Krishnasamy,[3] Mohammad Mazyad Hazzazi,[4] Prashant Kumar Shukla,[5] Piyush Kumar Shukla,[4,6] and Stephen Jeswinde Nuagah[7]

[1]*Department of Information Science and Engineering, AMC Engineering College, Bangaluru, India*
[2]*Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India*
[3]*Department of Information Technology, Kongu Engineering College, Tamil Nadu, India*
[4]*Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia*
[5]*Department of Computer Science and Engineering, K L University, 29-36-38, Museum Rd, Governor Peta, Vijayawada, Andhra Pradesh 520002, India*
[6]*Computer Science & Engineering Department, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, (Technological University of Madhya Pradesh), Bhopal 462033, India*
[7]*Department of Electrical Engineering, Tamale Technical University, Ghana*

Correspondence should be addressed to Stephen Jeswinde Nuagah; jeswinde@tatu.edu.gh

Received 19 September 2021; Revised 28 October 2021; Accepted 19 November 2021; Published 14 December 2021

Academic Editor: Paolo Barsocchi

The Internet of Things (IoT) is enhancing our lives in a variety of structures, which consists of smarter cities, agribusiness, and e-healthcare, among others. Even though the Internet of Things has many features with the consumer Internet of Things, the open nature of smart devices and their worldwide connection make IoT networks vulnerable to a variety of assaults. Several approaches focused on attack detection in Internet of Things devices, which has the longest calculation times and the lowest accuracy issues. It is proposed in this paper that an attack detection framework for Internet of Things devices, based on the DWU-ODBN method, be developed to alleviate the existing problems. At the end of the process, the proposed method is used to identify the source of the assault. It comprises steps such as preprocessing, feature extraction, feature selection, and classification to identify the source of the attack. A random oversampler is used to preprocess the input data by dealing with NaN values, categorical features, missing values, and unbalanced datasets before being used to deal with the imbalanced dataset. When the data has been preprocessed, it is then sent to the MAD Median-KS test method, which is used to extract features from the dataset. To categorize the data into attack and nonattack categories, the features are classified using the dual weight updation-based optimal deep belief network (DWU-ODBN) classification technique, which is explained in more detail below. According to the results of the experimental assessment, the proposed approach outperforms existing methods in terms of detecting intrusions and assaults. The proposed work achieves 77 seconds to achieve the attack detection with an accuracy rate of 98.1%.

1. Introduction

Considering the rapid development of IoT (Internet of Things) [1] technology, researchers and developers are urged to look at new smart services that can extract vital information from IoT data [2]. When physical objects such as mobile devices, home appliances, vehicles, and buildings are implanted with electronics, software, sensors, and network connectivity, the Internet of Things (IoT) came into effect. The Internet of Things (IoT) enables these objects to collect and exchange data with one another. By using the existing network infrastructure, the Internet of Things allows things to be detected, identified, and controlled from a distance. When the Internet of Things is combined with sensors and actuators, it becomes an example of cyber-physical systems, which also incorporate concepts such as intelligent grids, intelligent homes, intelligent cities, and intelligent transportation systems. The IoT has developed as a highly technical area in the current scenario [3]. In many applications, IoT devices have been implemented, including intelligent vehicles, healthcare, environmental monitoring, and personal wearable devices which have increased the volume, variety, speed, and veracity of data that are handled with connected systems (including sensitive data like personal information) [4–11]. The Internet of Things (IoT) builds on the idea of expanding everyday activities via computing and Internet connections to detect, compute, communicate, and control the environment [5, 6]. Along with the growth of IoT ecosystems, the extent and intensity of adverse actions have increased, focused on the resilience of the system, and thus affected latency-sensitive applications [6]. The poor physical safety of the devices [8] enables inside attacks against IoT nodes. It provides useful suggestions about the safety of industrial IoT systems [9]. As attackers tried to reveal the integrity of data and equipment [10], security issues emerged because of the rapid growth of hacking techniques. However, there will be data anomalies [11] and attacks like man-in-the-middle, message changes, authentication attacks, denial-of-service attacks, replay attacks, and eavesdropping which may threaten the viability of the Internet of Things [12]. The bulk of these attacks are slight differences in previously reported incursions (around 99 percent mutations). Security and privacy problems must be solved before IoT devices can be extensively utilized, especially in mission-critical scenarios involving sensitive data [13]. Anomalies or outliers are uncommon, but they can still be important in the event of a credit card transaction; for example, the aberrant conduct of credit card transactions may suggest that a credit card is stolen, whereas strange network traffic patterns can indicate that illegal network access is available [14]. According to the 2019 threat report SonicWALL, in 2018 in Gatlan (2019), more than 327 million threats from the Internet of Things (IoT) were detected globally [15]. As a consequence, assaults on and detection of abnormalities in IoT infrastructure in the IoT industry become an increasing cause of concern [16].

In the past, a wide variety of techniques for attack detection were proposed with various scenarios [17]. Computer algorithms based on artificial intelligence (AI) methods and machine learning (ML) have been extensively used to detect anomalous patterns of traffic in the network. While some excellent work has been done on the security issues related to the Internet of Things [18], many solutions designed to prevent security threats cannot be able to protect from new attacks [19]. Consequently, new kinds of attacks or novel variants in current attacks might not be identified [20]. In this paper, an effective anomaly detection system based on DWU-ODBN is proposed to solve the issues that presently exist. The fundamental goal of this work is to develop a new architecture for the IoTto defeat energy-related threats. A denial-of-sleep attack occurs when the nodes remain awake even when there is no traffic, resulting in the battery's energy being depleted and the node dying as a result of the assault. The lifespan of the IoT will be reduced as a result of this action. The primary goal of this research endeavour is to develop effective defence approaches for identifying and defending against DDoS attacks to prevent them from occurring. The following are the objectives: A review of different IDS in WSN and MANET, identify the steps for launching a denial-of-service attack, and develop and deploy an innovative technique for identifying and isolating rogue agents. Intruders from inside the network are responsible for the energy drain assaults. Aa DWU-ODBN with intrusion detection and prevention was designed and implemented. Attack prevention is made possible by the introduction of a new malicious node alert (MNA). Nodes in the network are alerted using this strategy. It is necessary to build a method that comprises a lightweight monitoring system module and detector module for detecting and preventing sleep deprivation in IoT.

The rest of the paper is organized as follows: Section 3 of this article explains the proposed approach, Section 4 of the paper includes a test assessment of the proposed methodology, and Section 5 of the paper ends the study with future improvements.

2. Literature Survey

Based on the SDx paradigm, Zarei and Fotohi [21] presented a comprehensive framework for the internet of things software-defined (SD-IoT). In addition to the SD-IoT controller package, the frame contained SD-IoT switches coupled to an IoT gateway and IoT devices connected via an IoT gateway. Then, an algorithm for detecting and mitigating DDoS attacks was created utilizing the SD-IoT platform. The cosine similarity of packet-in message rate vectors at border SD-IoT switch ports has been used to evaluate if DDoS assaults have occurred in the IoT. Last but not least, test results showed that the algorithm provided was very good and the framework offered was changed to enhance security in the Internet of Things while dealing with a variety of vulnerable devices. Since it was intended to be used only with directed vectors, the cosine similarity function could not be utilized for normal values.

Liu et al. [22] presented a new safe things architecture internet, in which the ensemble machine learning technique can detect IoT sensor risks based on their results. In an Internet of Things environment, the gradient boosting technique was selected with minimum modifying hyper-parameters and applied to identify discriminant attacks. In total, the system included a variety of processes, including internet data collection of the things sensor networks (IoT sensor networks), data clearing (data visualization), vectorization, ensemble learning (ensemble learning), and attack detection. First, the feature selection method was utilized to decrease the dataset size, which improved the attack and detection environment. The feature selection method was then utilized to decrease the dataset size. To get the best results, the ensemble gradient boosting technique was employed after the previous stage with only a little hyperparameter adjusting. Attacks on the environment of the IoT sensor were more correctly identified by the model with an accuracy of 99.40%, precision of 99%, and an F1 scoring of 99%, which showed that the model was more efficient. The ensemble method utilized throughout the process required more time to train for learning.

The novel lightweight random neural network (RaNN) prediction model has been suggested according to Latif et al. [23]. The collection and monitoring of datasets were the first stages in developing this architecture. The dataset was compiled and assessed according to its data type at this point. Dataset preprocessing was performed in the following step, including data cleansing, visualization, feature engineering, and vectorization. All of these techniques have been utilized to extract data from the data collection. These characteristic vectors were separated into two groups: a training set and a test set, where the two sets split 80 percent to 20 percent. The recommended random neural network was utilized to study the training set employed along with the proposed random neural network. To investigate RaNN-based prediction model performance, a variety of evaluation parameters were calculated and compared with traditional artificial neural network (ANN) performance, the support vector system (SVM), and decision-tab prediction models (DT). The results of this evaluation showed that the proposed RaNN model achieved a 99.20-percent accuracy with a learning rate of 0.01 and a prediction time of 34.51 milliseconds while training with a learning rate of 0.01. The precision of the system was hindered by the random character of the functioning of the network.

Chang et al. [24] presented a three-hierarchy joint local-global anomaly detection framework (HADIoT) that provides Internet of Things devices with sensory information generated and transferred to the local edge servers for local data anomaly recognition including data framing, standardization, analysis of principal components, and symbol mapping. The gated recurrent unit was used to focus local AD on the data consistency of certain devices and then sent the processed data from the edge servers for global AD to the cloud server. In the analysis of data pattern correlations between different Internet of Things devices, which were the focus of global AD, the conditional random fields were used. Simulations using a real dataset, the 2012 Information Security Center of Excellence (ISCX) dataset, were also utilized to evaluate the proposed method's performance for the study. Compared to three benchmarking schemes, the results of the simulation indicated that the proposed framework has been more effective than the benchmarks in terms of true positive rates, false positive rates, precision, exactness, and *F* score. Due to the low learning efficiency of the GRU, a longer training time was necessary.

Guo et al. [25] investigated uncontrolled anomaly detection on Internet multidimensional time series data of things systems and developed a Gaussian GRU-based VAE mixture technique which, in brief, was called GGM-VAE. Specifically utilized to identify correlations between time series data include the gated recurrent unit (GRU), whereas Gaussian mixture priors were used in latent space to characterize multimodal data, which is subsequently used to find correlations between time series data. During the training phase, further development was undertaken under the Bayesian inference criterion (BIC) to select the model best suited to estimate the latent distribution of the Gaussian mixture. According to the results of extensive simulations carried out on four datasets, the approach detected the latest abnormality. Due to the lack of flexibility in the Gaussian model mixing, many calculation errors occurred when creating the multimodal data.

3. Proposed Anomaly Detection System Model

Every security vulnerability of different Internet of Things devices that has been deployed in real time may be very difficult to detect and address. This is because many Internet of Things devices are designed with little attention to their safety implications. It is thus important to create ways to prevent these dangers or attacks and to protect IoT devices from malfunctioning or failure. Current IDS and other attack detection methods could not detect attacks that were started dynamically or during the online procedure. To solve the attack issues, efficient secure anomaly detection has been proposed for an IoT context, as detailed in the article. The proposed system is split into four phases: preprocessing, feature extraction, selection of features, and classification. The first of these procedures is preprocessing. Initially, the input data is handled using four steps called NaN value handling, categorical features, missing values, and an imbalanced dataset. The random sampling method is used to handle the imbalanced dataset. Then, the most important features are chosen from the preprocessed data and compared to the originals using the median absolute deviations in the mid-based Kolmogorov-Smirnov test (MAD Median-KS test). The most important properties are then selected from the gathered functions, using the optimization technique based on robust confidence interval chimp, which is explained in full below (RCI-ChOA). Once this is done, we will apply the selected features to the optimum deep faith network based on dual weight update (DWU-ODBN) to translate the categorized output into the attack and nonattack data. Figure 1 shows a block diagram with each block indicating a step.

3.1. Preprocessing. Initially, the data is preprocessed, which includes handling of NAN values, categorical features, unbalanced datasets, and missing values, which occurs both randomly and nonrandomly. After that, the data is processed further. The preprocessing step assists us in obtaining healthier data while also reducing the difficulties associated with the data, which inhibits the flow of data traffic. The following are the stages involved in the preprocessing process.

In handling NaN values, NaN is an abbreviation that stands for "Not a Number," and it is one of the most often used symbols to indicate a missing value in data. To improve the accuracy of attack detection, the input data for an attack detection system must be devoid of NaN values. The following is the procedure for dealing with NAN values in the input data:

$$d_n^{\mathrm{handl(NaN)}} = \psi_{\mathrm{hand\_NaN}}(d_n), \tag{1}$$

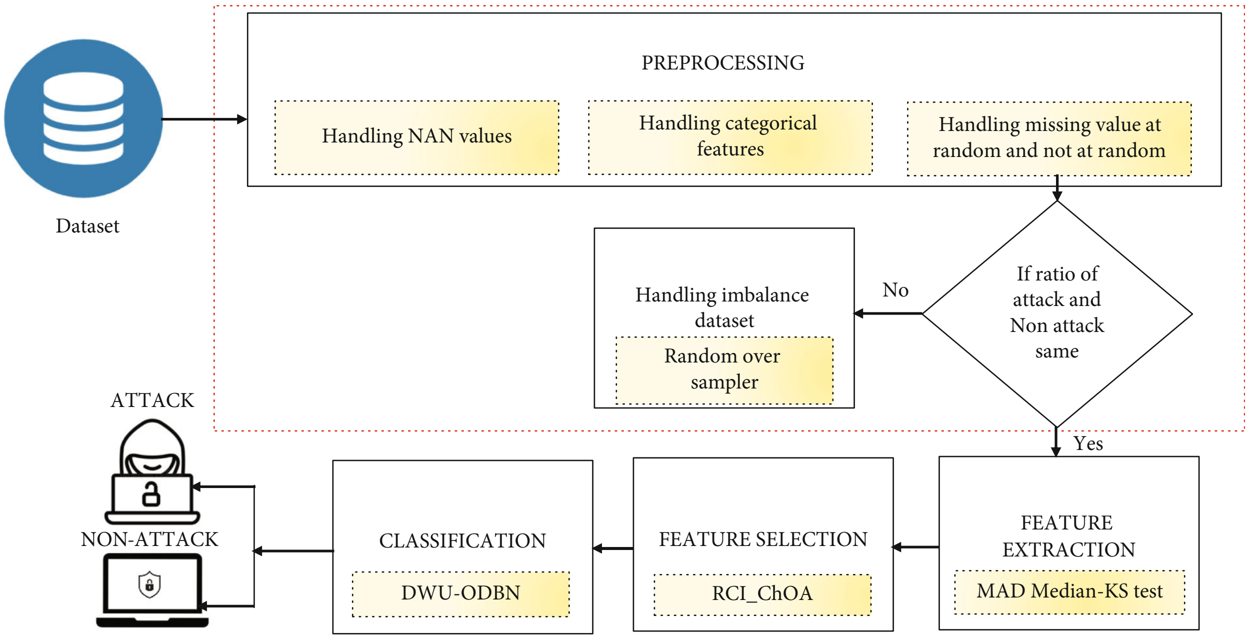

Figure 1: Block diagram of the proposed methodology.

where d_n is the input data, $\psi_{\text{hand\_NaN}}$ is the method that handles the NaN values, and $d_n^{\text{handl(NaN)}}$ is the data obtained after handling of NaN values.

In handling categorical features, handling categorical characteristics comes next once dealing with NaN values has been completed successfully. Categorical data must be processed in this phase before it can be fed into the machine learning models, which is the last step. Because machine learning models are considered mathematical models, they will not function properly with data that is stored in the texture format. The categorical characteristics may be handled in the following ways:

$$d_n^{\text{handl(ctg)}} = \xi_{\text{hand\_ctg}}(d_n), \tag{2}$$

where $\xi_{\text{hand\_ctg}}$ denotes the method that handles the categorical data values and $d_n^{\text{handl(ctg)}}$ is the data obtained after handling of categories.

In handling of missing values, in this step, the missing value at random and the missing values not at random are handled. Missing the values from some subsamples of data is considered as missing values at random. If the missing data has a specific structure, then it is known as missing values, not at random. The missing values can be handled as

$$d_n^{\text{handl(mis\_val)}} = \zeta_{\text{hand\_mis}}^{(\text{ran,not-ran})}(d_n), \tag{3}$$

where $\zeta_{\text{hand\_mis}}^{(\text{ran,not-ran})}$ is the method that handles the missing values and $d_n^{\text{handl(mis\_val)}}$ is the data obtained after handling of missing values.

Thus, the final set of preprocessed data is obtained as

$$d_n^{\text{pp}} = \{d_1, d_2, d_3, \cdots \cdots \cdots d_N\}, \tag{4}$$

where d_n^{pp} denotes the preprocessed data and N is the number of data. Then, the ratio of attacked and nonattacked data is checked to get the balanced and imbalanced data as

$$d_n^{\text{pp}} = \begin{cases} d_n^{\text{bal}} & \text{if}\left(A\left(d_n^{\text{pp}}\right) = \text{NA}\left(d_n^{\text{pp}}\right)\right), \\ d_n^{\text{imbal}} & \text{if}\left(A\left(d_n^{\text{pp}}\right) \neq \text{NA}\left(d_n^{\text{pp}}\right)\right), \end{cases} \tag{5}$$

where d_n^{bal} shows the balanced data, d_n^{imbal} shows the imbalanced data, A describes the attacked data, and NA describes the nonattacked data. From the obtained data, the balanced data is contributed to the next phase and the imbalanced data is handled using a random over sampler.

In handling of imbalanced dataset, here, the imbalanced data d_n^{imbal} is balanced with the help of a random oversampler. The random oversampler produces the balanced data using arbitrarily repeating the instances from the minority class and extreme them to the exercise input. The oversampling of data is

$$d_n^{\text{imbal}} \xrightarrow{\text{oversampling}} d_n^{\text{bal}}. \tag{6}$$

After that, the balanced data obtained with the help of oversampling is subjected to the feature extraction phase.

3.2. Feature Extraction.

3.2. Feature Extraction. Following that, to gain knowledge of the balanced data, the work has developed a median absolute deviation around the median-based Kolmogorov-Smirnov test (MAD Median-KS test) that passages techniques from the net and uses arithmetical examination to detect abnormalities originating from compromised IoT devices. The MAD Median-KS test takes into account the distribution similarity and eliminates the repeated distribution data, and it applies to both continuous and discrete data. It is important to note that the current Kolmogorov-Smirnov test only focuses on the highest difference value, which is affected by outliers and decreases the accuracy of identifying the attack. To get around this, the researchers utilized median absolute deviations around the median in their research.

In the MAD median-KS test, the features for the input data is computed as

$$\sigma_{\mathrm{ks}} = \alpha.\Lambda_n \left| \partial\left(d_n^{\mathrm{bal}}\right) - \Lambda_i\left(\Gamma\left(d_n^{\mathrm{bal}}\right)\right) \right|, \tag{7}$$

where α is the constant that disregards the abnormality induced by the outliers, Λ_n denotes the median of the distribution functions, ∂, Γ are the two distribution functions of the sample d_n^{bal}, Λ_i is the median of Γ which observes the features of the data. The extracted features such as duration, protocol type, service, flag, root_shell, count, server_count, and etcare denoted as

$$d_n^f = \left\{ d_1^{f(1)}, d_1^{f(2)}, d_1^{f(3)}, . \cdots \cdots \cdots d_n^{f(k)} \right\}, \tag{8}$$

where f denotes the number of features extracted from the input data and $f(k)$ denotes the k^{th} feature of n^{th} data.

3.3. Feature Selection. To maintain a superior computational speed and accuracy, the robust confidence interval-based chimp optimization method is used to extract and select important features after feature extraction (RCI-ChOA). The random updation of parameters in the existing chimp optimization algorithm leads to inaccurate model driving and chasing of prey, which results in the selection of features that are not relevant to the problem; therefore, to overcome this problem, the work has used robust confidence interval to update the parameters.

The ChOA algorithm simulates the social behavior of individual intelligence as well as the sexual drive of chimpanzees throughout a hunting expedition.

According to the retrieved characteristics, the chimp population that lives in a fission-fusion society is divided into four groups: (a) drivers, (b) barriers, (c) chasers, and (d) attackers (see Figure 1). This procedure is divided into two phases: the exploration stage and the exploitation stage, which are both important stages in the chimpanzees' hunting process. The exploration stage consists of driving, blocking, and pursuing, while the exploitation stage consists of assaulting the target and completing the mission.

When driving, the drivers simply follow the prey and do not make any effort to capture them. To impede the prey's entrance, the chimpanzees took up positions in trees and obstructed the prey's path. If you are pursuing something, you are trying to catch that something. In exploitation, the attackers [26] forecast the most advantageous path to take to assault the victim.

The driving and chasing of the prey can be modeled mathematically as

$$R = \left| \nu.\chi_p(q) - \vartheta.\chi_c(q) \right|, \tag{9}$$

$$\chi_c(q+1) = \chi_p(q) - \upsilon.R, \tag{10}$$

where ν and υ are the coefficient vectors $\nu = 2\varphi_2$, ϑ is the chaotic vector computed concerning various chaotic maps, and $\chi_p(q)$ and $\chi_c(q)$ are the position of the prey and a chimp at the number of current iterations q. The chaotic vector ϑ expresses the effect of the sexual motivation of chimps during the hunting process. The coefficient vectors are calculated as

$$\upsilon = \mathfrak{I}(2\varphi_1 - 1), \tag{11}$$

$$\nu = 2\varphi_2, \tag{12}$$

where $\mathfrak{I}$ is the parameter that reduces nonlinearly from 2.5 to 0 in both exploration and exploitation stages and φ_1, φ_2 are the parameters updated using robust confidence intervals $\mathrm{Con.Int}(H, F)$. The robust confidence intervals are calculated as

$$\mathrm{Con.Int}(H, F) = \mathrm{med} \pm \varepsilon_{(\gamma/2, n-1)} \frac{\sigma}{\sqrt{n}}, \tag{13}$$

where γ denotes the confidence coefficient, ε is the percentage point of the confidence interval, σ is the sample standard deviation, and med denotes the sample median. Then, the fitness is evaluated based on the computation time and accuracy of the system.

For attacking the prey, the attackers find the position of the prey with the help of a driver, barrier, and chasers. The chimps also attack the prey based on chaotic strategy. Here, two approaches are employed as exploring the location of the prey and encircling the prey. Since the initial position of the prey was unknown, the attacker, driver, barrier, and chaser update the position of the prey. The position updation can be expressed as

$$R_{\mathrm{attacker}} = |\nu_1 * \chi_A - \vartheta_1 * \chi|, \tag{14}$$

$$R_{\mathrm{barrier}} = |\nu_3 * \chi_B - \vartheta_3 * \chi|, \tag{15}$$

$$R_{\mathrm{chaser}} = |\nu_2 * \chi_{Ch} - \vartheta_2 * \chi|, \tag{16}$$

$$R_{\mathrm{driver}} = |\nu_4 * \chi_D - \vartheta_4 * \chi|. \tag{17}$$

The four optimal solutions are stored, and other chimps are required to update their solutions. If the random vectors of ν lies in [1,–1], then the next position of the chimp can be at any location in the middle of its current position and the position of prey. The location of the chimp can be updated as

$$\chi(1) = \chi_A - \upsilon_1.R_{\mathrm{attacker}}, \tag{18}$$

$$\chi(2) = \chi_{\mathrm{Ch}} - \upsilon_2.R_{\mathrm{chaser}}, \tag{19}$$

$$\chi(3) = \chi_B - \upsilon_3.R_{\mathrm{barrier}}, \tag{20}$$

$$\chi(4) = \chi_D - \upsilon_4.R_{\mathrm{driver}}. \tag{21}$$

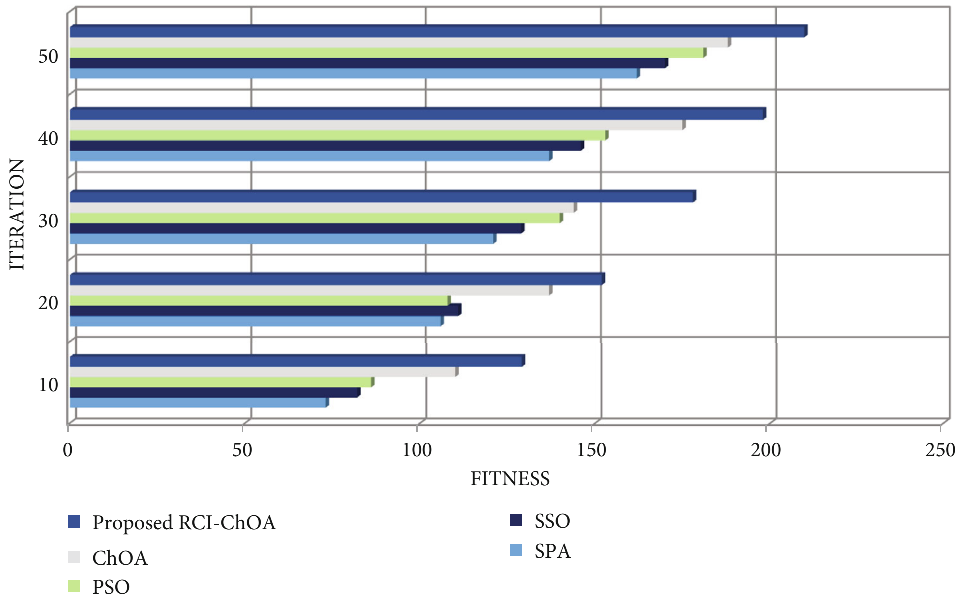

Figure 2: The performance of proposed RCI-ChOA based on fitness vs. iteration.

From the obtained locations, the position of the Chimp can be updated as

$$\chi(q+1) = \frac{\chi(1)+\chi(2)+\chi(3)+\chi(4)}{4}. \tag{22}$$

The hunting process of chimps can be affected by social benefits like support and sex. This motivates chimps to forget their contribution to the hunting process. So, the chaotic maps are used in the final stage of attacking the prey which helps chimps to lessen the local optima and slow convergence rate problems. The probability of choosing the normal position updation and chaotic map-based position updation is determined by the parameter $\eta \in (0,1)$ as

$$\chi_c(q+1) = \begin{cases} \chi_p(q+1) - \vartheta.R, & \text{if}(\eta < 0.5), \\ \varsigma_{m(\text{pos})}, & \text{if}(\eta > 0.5), \end{cases} \tag{23}$$

where ς_m denotes the chaotic map-based position updation process. Based on the updated position, the prey has been attacked by the attackers [27]. The hunting process will be stopped when the movement of the prey was stopped. The pseudocode of the RCI-ChOA method is shown in Figure 2,

Algorithm 1 explains the steps involved in the RCI-ChOA method. In the same way as the hunting behavior of the chimps, the optimal features $d_n^{f_{\text{opt}}}$ are selected from the extracted features.

3.4. Classification. The last stage involves training and evaluating the selected features over a dual weight updation-based optimal deep belief network (DWU-ODBN) classification system to detect and classify attacks [28]. Semisupervised classification algorithms, in contrast to supervised classification techniques, are capable of coping with adversarial attacks as well as uncertain attacks, which makes them more robust. As a result of the technique's low error rate as well as its low false alarm rate (FAR), it can detect assaults more quickly and reliably. There are two RBM layers in the DBN, each of which consists of a single observable coating and one concealed coating. There are three RBM layers in the DBN. MLP is composed of three layers: the input layer, a hidden layer, and a final output layer. The input layer is the smallest of the three layers, and it is composed of the hidden layer and the final output layer. Only between the visible layer neurons and the hidden layer neurons does the connection form in DBN; the link does not form between the visible layer neurons and the hidden layer neurons in other words. However, there is no joining amid the visible nerve cell in the brain and the hidden neurons in other parts of the body.

Dual weight updation is used to enhance the accuracy of the classification process while simultaneously reducing the error rate of the classification process. Examples of weight updates in the proposed classification technique include stochastic gradient descent with momentum (SGDM), which has a higher convergent rate than gradient descent, and weight updates in the proposed classification technique include the best neighbor direction-based seagull optimization algorithm, which has a higher convergent rate than gradient descent (BND-SOA). Due to inaccuracy in the existing seagull optimization, the construction of the best neighbor direction results in high variance and standard deviation, which may result in poor convergence for updating weight as well as a long computation time for the update weight to be done. To overcome this barrier, the best neighbor direction based on Euclidean distance is determined.

The selected characteristics are presented as the input to the visible layer of the first RBM in DBN, which is the visible layer of the first RBM at the start of the simulation. The output of the visible layer is then mapped to the hidden layer of the first RBM, and the process is repeated until the output of

```
Input: Extracted features $d_n^f$
Output: Selected features $d_n^{f_{\text{opt}}}$
Begin
    Initialize population $d_n^f$, coefficient vectors $v, \vartheta, \upsilon$, maximum number of iteration $q_{\max}$
    Calculate fitness for each chimps
    Set$q = 0$
    While $(q \leq q_{\max})$
        Update $v, \vartheta, \upsilon$
        Update position using $R_{\text{attacker}}, R_{\text{chaser}}, R_{\text{barrier}}, R_{\text{driver}}$
        Evaluate fitness of the position of chimps
            If $(\mu < 0.5\&\&v \in (0, 1))$ {
                Update position of the chimps using $\chi_p(q+1) - \vartheta.R$
            Else
            {
                Update position of the chimps using $\varsigma_{m_{\text{pos}}}$
            } End if
        Update $v, \vartheta, \upsilon$
        Calculate fitness of the current position of the chimp
    Set $q = q + 1$
    End while
    Return selected features $d_n^{f_{\text{opt}}}$
End
```

Algorithm 1: Pseudocode of the RCI-ChOA method

the visible layer is no longer visible. These modifications to RBM weights are accomplished via the use of SGDM.

$$\varpi_i = \lambda\varpi_{i-1} + (1-\lambda)G^{(-i)}, \tag{24}$$

$$h^{i+1} = h^i - \tau\varpi_i, \tag{25}$$

where λ is the momentum coefficient, τ is the step size, h^{i+1} is the intermediate model parameter, and $G^{(-i)}$ is the gradient of the weight values. The output of the hidden layer is expressed as

$$\Phi^1_{\text{hid\_out}} = \mathfrak{R}\left(B^1_s + \sum_i \Phi^1_{\text{vis\_out}}\varpi^1_{i(\text{vis\_hid})}\right), \tag{26}$$

where $\mathfrak{R}$ is the activation function, B^1_s represents the bias of the hidden units, $\Phi^1_{\text{vis\_out}}$ is the output of the visible layer, and $\varpi^1_{i(\text{vis\_hid})}$ is the weight values between the visible and hidden neurons using the SGDM method.

$$\Phi^2_{\text{hid\_out}} = \mathfrak{R}\left(B^2_s + \sum_i \Phi^1_{\text{hid\_out}}\varpi^2_{i(\text{vis\_hid})}\right), \tag{27}$$

where $\varpi_{\text{hid\_vis}}$ is the weight values between the visible and hidden neurons of the second RBM, $\Phi^1_{\text{hid\_out}}$ is the output from the previous RBM, and B^2_s denotes the bias of the hidden units. Then, the output obtained from the second RBM is inputted to the MLP.

The MLP contains two weight vectors between the input and hidden layer and amid the hidden and output layer. In the MLP layer, the optimal weight values are updated using the BND-SOA method. The SOA method is the inspiration for migrating and attacking behaviors of the seagulls. During migration, a group of seagulls shifts their position to another position. In this migration process, it needs to consider three conditions which are detailed below:

(i) Collision avoidance: to avoid collision between the search agents, the position of the new search agent is computed as

$$D_{\text{SA}} = E \times C^p_{\text{SA}}(z), \tag{28}$$

where E is the variable that represents the search agent's movement behavior and reduces from its maximum frequency to zero, C^p_{SA} denotes the current position of new search agent, and D_{SA} is the position of the search agents which does not involve in collision with other search agents

(ii) The movement regards the best neighbor's direction: once the collision has been avoided, the search agents move by facing the direction of the best search agent. The movement of the search agents is expressed using the Euclidian distance as

$$\text{Mt}_{\text{SA}} = 2 \times E^2 \times Q\sqrt{\sum_{z=1}^{z_{\max}}\left(C^p_{\text{BSA}(z)} - C^p_{\text{SA}(z)}\right)^2}, \tag{29}$$

where $Q \in (0, 1)$ is the random parameter and $C^{p}_{\mathrm{SA}(z)}$ is the position of search agent towards the position $\mathrm{Mt}_{\mathrm{SA}}$ of the best search agent $C^{p}_{\mathrm{BSA}(z)}$ at z

(iii) *Endure close to the best search agent:* the position of the search agent is updated regarding the position of the best search agent as

$$L_{\mathrm{SA}} = \left| D_{\mathrm{SA}} + \mathrm{Mt}_{\mathrm{SA}} \right|, \tag{30}$$

where L_{SA} is the distance between D_{SA} and $\mathrm{Mt}_{\mathrm{SA}}$.

After that, in the attacking action, the seagulls change the speed and angle of attack continuously. The change can be done using their weights and wings and makes spiral movement behavior in the air. The spiral movement behavior is defined in Xx, Yy, and Zz plane as

$$Xx = \mathrm{rad} \times \cos (u), \tag{31}$$

$$Yy = \mathrm{rad} \times \sin (u), \tag{32}$$

$$Zz = \mathrm{rad} \times u, \tag{33}$$

where each turn in the spiral contains a certain radius and is denoted as rad and $u \in (0, 2\pi)$ is the random number. Then, the fitness is evaluated based on the error rate of the MLP layer. The updated position can be calculated using the spiraling action as

$$C^{p}_{\mathrm{SA}}(z) = \left(L_{\mathrm{SA}} \times Xx \times Yy \times Zz \right) + C^{p}_{\mathrm{BSA}}(z), \tag{34}$$

where $C^{p}_{\mathrm{SA}}(z)$ is responsible for storing the best position of the search agents and updating the position of other search agents. In this way, the weight values for the MLP layer $\varpi^{\mathrm{MLP}}_{i}$ are selected.

Concerning the input values, the output of the hidden layer in MLP is denoted as

$$\Phi^{\mathrm{MLP}}_{\mathrm{hid\_out}} = \left(\sum_{i=1}^{K} \Phi^{2}_{\mathrm{hid\_out}} \varpi^{\mathrm{MLP}}_{i(\mathrm{ip\_hid})} \right) B^{\mathrm{MLP}}_{s}, \tag{35}$$

where $\varpi^{\mathrm{MLP}}_{i(\mathrm{ip\_hid})}$ is the weight value between the input and hidden layers of the MLP, K is the number of neurons in the hidden layer, and B^{2}_{s} is the bias of the hidden neuron in MLP. Based on the output of the hidden layer, the output vector is calculated as

$$\Phi^{\mathrm{MLP}}_{\mathrm{op\_out}} = \left(\sum_{k=1}^{K} \varpi^{\mathrm{MLP}}_{k(\mathrm{hid\_out})} \Phi^{\mathrm{MLP}}_{\mathrm{hid\_out}} \right) B_{i}, \tag{36}$$

where $\varpi^{\mathrm{MLP}}_{k(\mathrm{hid\_out})}$ is the weight value between the hidden and output layer. The output unit contains two classes of data as attack and nonattack. The proposed system detects adversarial attacks and uncertain attacks as well. Adversarial attacks mean the data are modified in such a way that, they cannot be detectable to the human eye.

Table 1: Evaluation of proposed RCI-ChOA based on fitness vs. iteration.

Techniques/iteration	10	20	30	40	50
SPA	73	106	121	137	162
SSO	82	111	129	146	170
PSO	86	108	140	153	181
ChOA	110	137	144	175	188
Proposed RCI-ChOA	129	152	178	198	210

4. Results and Discussion

In this part, the efficacy of the suggested irregularity discovery outline for use in the Net of Belongings setting is assessed using case studies. The experiments for assessing the suggested system are carried out using the Python programming language's working platform.

4.1. Database Description. The NSL-KDDCUP99 dataset is utilized in this study for experimental assessment. KDD Cup'99 is a dataset that is often used in the development of intrusion detection systems. Per network connection, the dataset includes a total of 41 characteristics. The characteristics are divided into several categories. Eighty percent of the data in the dataset is utilized for training, whereas only twenty percent of the data is used for testing.

4.2. Performance Evaluation of Feature Selection Technique. Based on the fitness vs. iteration trade-off, this section evaluates the performance of the proposed RCI-ChOA technique. The suggested technique is compared to the current SPA, SSO, particle swarm optimization (PSO), and chimp optimization algorithm (ChOA) methods to determine its suitability for the task at hand.

In discussion, fitness versus iteration is used to assess the performance of the proposed and current techniques, as shown in Table 1. The number of iterations required for the analysis is shown in Table 1 and ranges from 10 to 50. The suggested technique has a fitness value of 129 for the 10 number iterations that are included in the analysis. For 10 iterations, the current SSO and PSO techniques have fitness values of 82 and 86, respectively, while the existing SPA and ChOA methods have fitness values of 73 and 110, respectively. When the suggested technique is evaluated for its fitness throughout the remaining number of iterations, it scores 152 for 20 iterations, 178 for 30 iterations, 198 for 40 iterations, and 210 for 50 iterations, according to the findings. According to the suggested approach, current methods have poorer fitness for the number of iterations in comparison to the new method. The discussion demonstrates that the suggested approach outperforms the current methods in terms of overall performance. Figure 2 depicts a graphical depiction of the fitness versus iteration analysis in terms of fitness,

4.3. Performance Evaluation of Classification Technique. The accuracy, sensitivity, specificity, precision, F measure, FPR, FNR, and MCC of the proposed DWU-ODBN technique are compared to the current ENN, CNN, SVM, and DBN

Table 2: Performance analysis of proposed DWU-ODBN based on quality metrics.

Performance metrics/techniques	ENN	CNN	SVM	DBN	Proposed DWU-ODBN
Accuracy	92.14	91.51	94.84	93.26	97.34
Specificity	86.67	87.57	85.82	93.39	93.42
Sensitivity	89.69	90.48	85.54	91.62	95.61
Precision	92.37	90.62	87.63	95.83	97.43
F measures	93.52	93.39	89.29	94.62	97.72
FPR	46.46	44.25	59.42	31.51	11.04
FNR	42.93	49.64	56.58	38.58	8.39
MCC	85.44	82.87	80.52	88.28	94.65

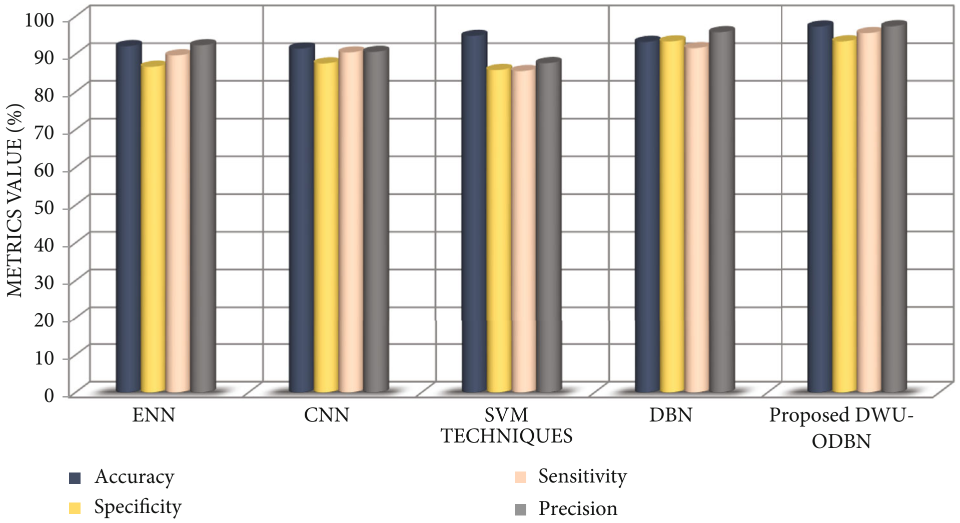

Figure 3: The performance analysis of the proposed DWU-ODBN based on accuracy, specificity, sensitivity, and precision.

methods in this section, as well as the existing ENN, CNN, SVM, and DBN methods. Following that, a comparison of the calculation time and attack detection time of the proposed and current techniques is provided.

In discussion, several quality indicators are used to assess the presence of the planned and current techniques, which are summarised in Table 2. For a classifier to be efficient, its accuracy, sensitivity, specificity, precision, *F* measure, and MCC values should all be greater than their current levels. The FPR and FNR are negative numbers that are often lower to achieve better categorization. As seen in the above table, the current SVM achieves low-level performance. Additionally, the current ENN, CNN, and DBN techniques have medium-level performance, according to the researchers. However, the suggested DWU-ODBN approach outperforms all other methods in terms of performance measures. In addition, it should be highlighted that the suggested approach exhibits improvement across all measures and outperforms the current ENN, CNN, SVM, and DBN methods in terms of overall performance.

In discussion, in the above Figure 3, the accuracy, sensitivity, specificity, and precision of the proposed and current techniques are evaluated about one another. The proposed method achieves classification accuracy of 97.34 percent, whereas the existing ENN, CNN, SVM, and DBN methods achieve accuracy of 92.14 percent, 91.51 percent, 94.84 percent, and 93.26 percent, respectively, which is lower than the proposed DWU-ODBN method. The proposed method outperforms the existing methods by a wide margin. The suggested approach also has better sensitivity, specificity, and precision than the current methods, with 95.61 percent, 93.42 percent, and 97.43 percent, respectively, compared to the existing methods. It is apparent from the above study that the suggested DWU-ODBN technique outdoes the other current methods in terms of overall performance.

In discussion, on the right, you can see how the proposed and current techniques compare in terms of *F* measure, FPR, FNR, and MCC in Figure 4. While the proposed method has an FPR of 11.04 percent and an FNR of 8.39 percent, existing ENN, CNN, SVM, and DBN methods have FPRs of 46.46 percent, 44.25 percent, 59.42 percent, and 31.51 percent, respectively, and FNRs of 42.93 percent, 49.64 percent, 56.58 percent, and 38.58 percent, respectively, which are higher than the proposed DWU-ODBN method. Similarly, the *F* measure and MCC of the suggested approach are 97.72 percent and 94.65 percent, respectively, which are greater than those of the current methods. According to the findings of the preceding study, the

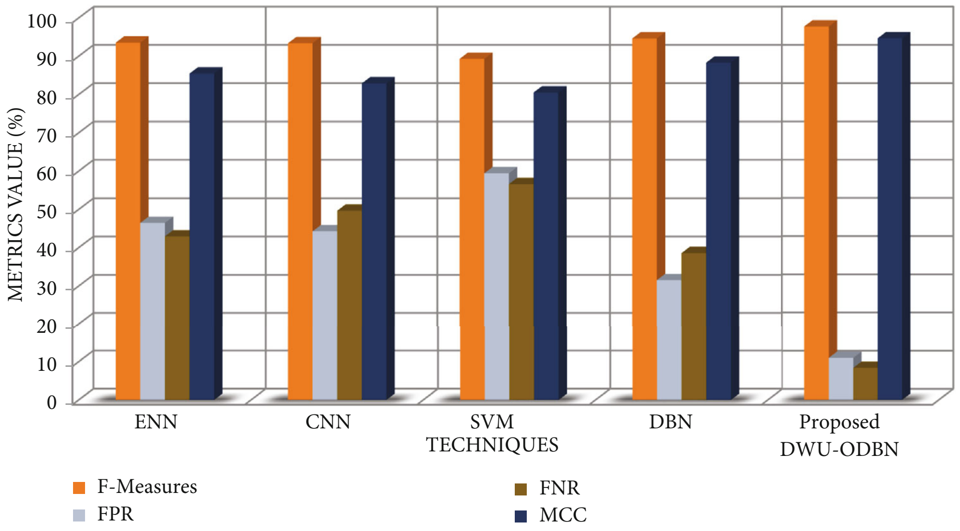

Figure 4: The performance of the proposed DWU-ODBN concerning *F* measure, FRR, FNR, and MCC.

suggested approach produces superior outcomes for all metrics than the current methods.

In discussion, Figure 5 compares the detection rates of the proposed and current techniques for detecting attacks. This study estimates the rate at which attacks are detected for two types of information: attack data and nonattack data. For the assault data, the proposed approach achieves a detection rate of 98.06 percent, while the current methods achieve a detection rate of 76.25 percent for ENN, 88.29 percent for CNN, 77.15 percent for SVM, and 86.36 percent for DBN, respectively. The suggested approach obtains a detection rate of 99.88 percent in the case of data that is not subjected to an assault. In comparison, the current ENN, CNN, SVM, and DBN techniques obtained rates of 77.76 percent, 87.56 percent, 78.65 percent, and 90.60 percent in the study, respectively. According to the results of the study, the suggested approach achieved the greatest detection rate for both attack data and nonattack data, whereas the detection rates of the current methods were much lower than the proposed method's detection rate. According to the results of the study, the suggested approach outperforms the current methods in terms of performance.

In discussion, Figure 6 compares the calculation times of the proposed DWU-ODBN technique with those of the current ENN, CNN, SVM, and DBN methods, among others. The study is carried out on several different datasets, including KDD99, NSL-KDDCUP99, CIDDS-001, and UNSW-NB15. The calculation time should be kept to a minimum for an efficient classifier. The suggested approach takes 77 seconds to compute for the KDD99 dataset, 78 seconds to compute for the NSL-KDDCU99 dataset, 71 seconds to compute for the CIDDS-001 dataset, and 62 seconds to compute for the UNSW-NB15 dataset. When compared to the suggested approach, the current methods need a longer calculation time for all datasets. The results of the study indicate that the suggested approach is more effective at detecting assaults than the already available techniques.

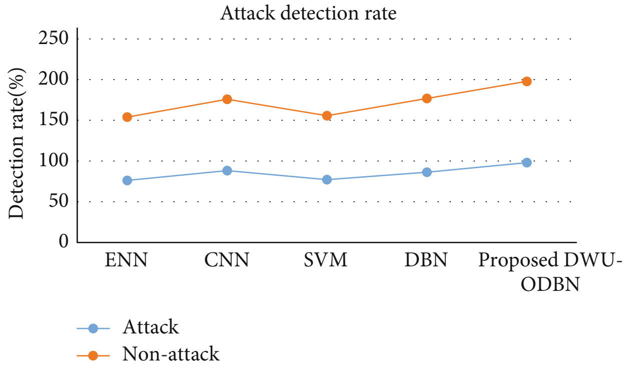

Figure 5: The attack detection rate of the proposed and existing methods.

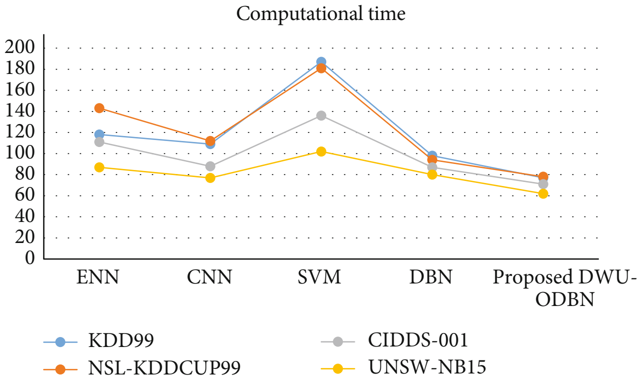

Figure 6: The computation time of the proposed and existing methods.

The result and analysis consider the feature selection, classification based on the performance metrics such as accuracy, sensitivity, specificity, precision, *F* measure, FPR, FNR, and MCC, attack detection rate, and computation time. These parameters were broadly compared against existing techniques. Further, the parameters would be designed based on the real-time scenario. This will be incorporated in the future work of this article.

5. Conclusion

Typically, IoT devices are resource-constrained, and those that do have the limited onboard capability for security operations may result in data breaches due to a lack of available functionality. In turn, it makes it more difficult for the system to detect cyberattacks from the network promptly before they impair smart city operations. This article presents a dual weight updating-based optimum deep belief network for attack detection to protect Internet of Things devices from a variety of security threats. The suggested technique is divided into four stages. It is essential to evaluate the presentation of the suggested system in command to use the KDD99 dataset. It is determined how well the suggested RCU-ChOA and DWU-ODBN techniques perform when compared to the already available methods. Although the suggested DWU-ODBN technique is more accurate than the current methods, the accuracy of the proposed method is greater than that of the existing methods. The suggested system has a calculation time of 77 seconds, which is much shorter than the current techniques. In this way, it is concluded that the suggested method outstrips the present approaches in footings of performance. The proposed system attains a 98% accuracy rate in detecting the attack from the malicious node. As we compared various dataset analyses through our proposed work, the proposed work detects the attack within 71 seconds. The proposed method achieves classification accuracy of 97.34 percent, whereas the existing ENN, CNN, SVM, and DBN methods achieve accuracy of 92.14 percent, 91.51 percent, 94.84 percent, and 93.26 percent, respectively, which is lower than the proposed DWU-ODBN method.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors of this manuscript declared that they do not have any conflict of interest.

Acknowledgments

King Khalid University Researchers Supporting Project Number (R. G. P. 2/150/42), King Khalid University, Saudi Arabia, supported this work.

References

[1] Y. Cheng, X. Yan, H. Zhong, and Y. Liu, "Leveraging semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in IoT communication," *IEEE Internet of Things Journal*, vol. 99, pp. 1–11, 2021.

[2] X. Rongbin, Y. Cheng, Z. Liu, Y. Xie, and Y. Yang, "Improved long short-term memory based anomaly detection with concept drift adaptive method for supporting IoT services," *Future Generation Computer Systems*, vol. 112, pp. 228–242, 2020.

[3] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks*, vol. 9, no. 3, pp. 120–127, 2020.

[4] G. De La Torre, P. R. Parra, K.-K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," *Journal of Network and Computer Applications*, vol. 163, p. 102662, 2020.

[5] P. Dymora and M. Mazurek, "Anomaly detection in IoT communication network based on spectral analysis and Hurst exponent," *Applied Sciences*, vol. 9, no. 24, pp. 5319-5320, 2019.

[6] A. Protogerou, S. Papadopoulos, A. Drosou, D. Tzovaras, and I. Refanidis, "A graph neural network method for distributed anomaly detection in IoT," *Evolving Systems*, vol. 12, no. 1, pp. 19–36, 2021.

[7] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 1–12, 2018.

[8] D. B. Gothawal and S. V. Nagaraj, "Anomaly-based intrusion detection system in RPL by applying stochastic and evolutionary game models over IoT environment," *Wireless Personal Communications*, vol. 110, no. 3, pp. 1323–1344, 2020.

[9] C. Wang, "IoT anomaly detection method in intelligent manufacturing industry based on trusted evaluation," *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 3, pp. 993–1005, 2020.

[10] M. Keshk, E. Sitnikova, N. Moustafa, H. Jiankun, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 1, pp. 66–79, 2021.

[11] W. Di, Z. Jiang, X. Xie, X. Wei, Y. Weiren, and R. Li, "LSTM learning with bayesian and gaussian processing for anomaly detection in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5244–5253, 2019.

[12] Z. A. Baig, S. Sanguanpong, S. N. Firdous, V. N. Vo, T. G. Nguyen, and C. So-In, "Averaged dependence estimators for DoS attack detection in IoT networks," *Future Generation Computer Systems*, vol. 102, pp. 198–209, 2020.

[13] A. Y. Khan, R. Latif, S. Latif, S. Tahir, G. Batool, and T. Saba, "Malicious insider attack detection in IoTs using data analytics," *IEEE Access*, vol. 8, pp. 11743–11753, 2019.

[14] I. Razzak, K. Zafar, M. Imran, and X. Guandong, "Randomized nonlinear one-class support vector machines with bounded loss function to detect of outliers for large scale IoT data," *Future Generation Computer Systems*, vol. 112, pp. 715–723, 2020.

[15] R. Vangipuram, R. K. Gunupudi, V. K. Puligadda, and J. Vinjamuri, "A machine learning approach for imputation and anomaly detection inIoTenvironment," *Expert Systems*, vol. 37, no. 5, pp. 1–16, 2020.

[16] M. Hasan, M. Milon Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019.

[17] D. H. Hoang and H. D. Nguyen, "A PCA-based method for IoT network traffic anomaly detection," in *International*

Conference on Advanced Communications Technology (ICACT), Chuncheon, Korea (South), 2018.

[18] Y. An, F. Richard YuJianqiang Li, J. Chen, and V. C. M. Leung, "Edge intelligence (EI)-enabled http anomaly detection framework for the internet of things (IoT)," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3554–3566, 2020.

[19] V. Kumar, A. K. Das, and D. Sinha, "UIDS a unifed intrusion detection system for IoT environment," *Evolutionary Intelligence*, vol. 14, no. 1, pp. 47–59, 2019.

[20] Y. Ebazadeh and R. Fotohi, "A reliable and secure method for network-layer attack discovery and elimination in mobile ad-hoc networks based on a probabilistic threshold," *Security and Privacy*, no. article e183, 2021.

[21] S. M. Zarei and R. Fotohi, "Defense against flooding attacks using probabilistic thresholds in the internet of things ecosystem," *Security and Privacy*, vol. 4, no. 3, article e152, 2021.

[22] Y. Liu, S. Garg, J. Nie et al., "Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[23] S. Latif, Z. Zou, Z. Idrees, and J. Ahma, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," *IEEE Access*, vol. 4, pp. 1–14, 2016.

[24] H. Chang, J. Feng, and C. Duan, "HADIoT a hierarchical anomaly detection framework for IoT," *IEEE Access*, vol. 4, pp. 1–10, 2016.

[25] Y. Guo, T. Ji, Q. Wang, Y. Lixing, G. Min, and P. Li, "Unsupervised anomaly detection in IoT systems for smart cities," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2231–2242, 2020.

[26] B. Butani, P. K. Shukla, and S. Silakari, "An exhaustive survey on physical node capture attack in WSN," *International Journal of Computer Applications*, vol. 95, no. 3, pp. 32–39, 2014.

[27] A. K. Saxena, S. Sinha, and P. Shukla, "Design and development of image security technique by using cryptography and steganography: a combine approach," *Graphics and Signal Processing (IJIGSP)*, vol. 10, no. 4, pp. 13–21, 2018.

[28] A. Kothari, P. Shukla, and R. Pandey, "Trust centric approach based on similarity in VANET," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 1923–1926, Paralakhemundi, India, 2016.