Peer-to-Peer Networking and Applications (2025) 18:79
https://doi.org/10.1007/s12083-024-01838-0

# Strengthening security in IoT-based smart cities utilizing cycle-consistent generative adversarial networks for attack detection and secure data transmission

**Agitha W[1] · D. R. Denslin Brabin[1] · K. Kalai Kumar[1] · T. Sunitha[2]**

Received: 6 September 2023 / Accepted: 11 October 2024 / Published online: 25 January 2025


**Abstract**
The main purpose of Smart Environments (SE) is to conveniently improve the human's daily life. Internet of Things (IoT) is a developing network for smart objects. Privacy-based security is a significant issue in any real-world smart environments centered on the IoT system. Security susceptibility in the IoT-centered systems provides a risk of security affecting smart environment applications. In this manuscript, Strengthening Security in IoT-Based Smart Cities utilizing Cycle-Consistent Generative Adversarial Networks for Attack Detection and Secure Data Transmission (IoT-SC-CCGAN-ADSDT) is proposed. Here, input information is gathered from NSL-KDD. The NSL-KDD input is pre-processed. Then, the important features of the pre-processed data are selected by using Wild horse optimizer (WHO). After feature selection, the chosen features are provided to cycle-consistent generative adversarial network classifier for classifying the attack and normal data. The selected features are sent to the use after the prediction of outcomes using Advanced Encryption Standard (AES). The AES is optimized using Chameleon Swarm Algorithm for transmitting the data in a safer way. After transmitting the data securely, the normal data outcomes obviously shown in LCD monitor. To show these results, major problems in the smart cities are simply detected. The proposed model is activated using java. The efficiency is examined with performance metrics, like precision, sensitivity, specificity, accuracy, computational time, encryption time, decryption time, security level. The proposed IoT-SC-CCGAN-ADSDT approach provides 96.68%, 7.142%, 94.65%, and 97.58% greater accuracy compared to the existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL methods respectively.

**Keywords** Advanced encryption standard · Attack detection · Chameleon swarm algorithm · Wild horse optimizer · Internet of Things · Smart cities

## 1 Introduction

IoT is a network of networked devices that facilitates the easy flow of information between devices, including industrial robots, medical equipment, smart home and environmental sensors, car and roadside sensors, surveillance devices [1, 2]. IoT has gained enormous popularity in communities and services globally in recent years. It is projected that there are expected to be over 125 billion linked IoT devices by 2030, up from 27 billion in 2017 [3, 4]. IoT devices use many services, technologies, and protocols. Future IoT infrastructure maintenance is consequently more challenging and ultimately increasing system vulnerability [5, 6]. IoT devices are used in applications for smart cities where cyber-attacks remotely alter device parameters making them insecure or obtain sensitive information about citizens' daily activities without the acquaintance of user (e.g., in Miria botnet attack) [7–10]. IoT platform assaults

✉ D. R. Denslin Brabin
denscse@gmail.com

Agitha W
ajithajerry@gmail.com

K. Kalai Kumar
kalaikumar23@gmail.com

T. Sunitha
sunithathangappan@gmail.com

[1] Department of Computer Science and Engineering, DMI College of Engineering, Chennai, Tamil Nadu, India

[2] Department of Artificial Intelligence and Data Science, Saveetha Engineering College, Chennai, Tamil Nadu, India

increased 600% in 2019, according to Symantec [11]. Attackers attempts to exploit the interconnection of the devices [12–14].

The proliferation of IoT devices in smart cities has brought about numerous advancements and conveniences. However, it has also introduced significant security challenges. IoT-based smart cities are susceptible to various kinds of cyber-attacks, like data breaches, spoofing, denial-of-service attacks. These attacks compromise the integrity, confidentiality, and critical data availability, posing serious risks to public safety and the efficient functioning of city services.

The study introduces a pioneering approach with the IoT-SC-CCGAN-ADSDT framework that innovatively combines Cycle-Consistent Generative Adversarial Networks (CCGANs) with an advanced optimization of AES utilizing the Chameleon Swarm Algorithm. This unique integration is novel in its dual focus on leveraging CCGANs for high-accuracy attack detection and enhancing data security through optimized encryption techniques. The originality of this method lies in its capability to address both cyber security and data protection challenges in IoT-based smart cities simultaneously, utilizing cutting-edge algorithms to deliver robust and secure solutions.

The major contributions of the proposed manuscript are abbreviated below:

- Proposes the IoT-SC-CCGAN-ADSDT framework, which uniquely integrates Cycle-Consistent GANs with AES optimized by the Chameleon Swarm Algorithm, offering a novel solution for attack detection and secure data transmission.
- The application of Chameleon Swarm Algorithm for optimizing AES, enhancing the safety of data transmission in IoT environments.
- CCGANs for sophisticated and accurate attack detection, setting a new standard in identifying and mitigating cyber threats in smart cities.
- Applies the NSL-KDD dataset to efficiently evaluate and validate the framework, encompassing diverse data types and attack scenarios.
- Performance of BTC-DCSNN–GEO is compared with the existing DL-IOT-SCA [15], IoT-SC-PCA [16], IoT-SCA-DL [17] models respectively.

Rest of the manuscript is arranged as: segment 2 presents literature survey, segment 3 describes proposed technique, segment 4 illustrates results, segment 5 gives conclusion.

## 2 Literature survey

Several researches were suggested depending on Deep Learning with Attack Detection in IoT Based SC. Some of the recent researches are revised below,

Rashid et al. [15] have presented Adversarial Training for Cyber-attack Detection in IoT-based Smart City Applications utilizing Deep Learning. The impact of adversarial assaults on deep learning with shallow machine learning was investigated using a current IoT dataset. The presented model substantially enhances IDS presentation if challenging adversarial attacks. The outcomes prove that the presented model attains detection accuracy in contradiction of total categories of attack involving adversarial attack. It provides minimum decryption time and greater computational time.

Alhanaya and Al-Shqeerat [16] have suggested developing an Integrated Framework for Protecting IoT Traffic in Smart Cities under Machine Learning methods to identify suspicious traffic and deal with secure data transmission in smart cities. The technique for creating an intrusion detection system (IDS) to find different forms of attacks was presented. It was performed by employing a Principal Component Analysis technique that minimizes system dimensionality and gets rid of redundancy. The presented model demonstrates employing an ensemble model that enhances performance of IDS. It provides higher specificity with maximum encryption time.

Hazman et al. [17] have presented Enhanced Intrusion Detection System using Deep Learning for IoT-Based Smart Cities Security The identification solution in this study provides interoperability with IoT connectivity standards, and smart IDS optimized for IoT threats was implemented. An ID was a network security technology that has received more attention from researchers recently. Scientists and industry have already expressed interest in the system's ability to detect breaches. Numerous IDSs rely on deep learning and machine learning was presented. The IDS-SIoDL, unique IDS that combines feature engineering with long short-term memory for IoT-based smart cities was presented. Tensor processing unit testing was performed on upgraded BoT-IoT, Edge-IIoT, NSL-KDD datasets to evaluate the model presented. It provides minimum decryption time and greater computational time.

Al-Farhani et al. [18] have suggested IoT with Blockchain-Based Cloud for Smart Cities Safe Data Transmission. The broad application of IoT technology offers both advantages and disadvantages. For the Internet of Things securely and prevent intrusions, a comprehensive and dependable security system must be in place. Many effective intrusion detection systems have been created as machine learning and deep learning technology have improved. Such two forms of security were compared. It attains greater specificity with maximal encryption time.

Rashid et al. [19] have suggested Cyber-attacks recognition in IoT-based smart city applications utilizing machine learning approaches. Examine a machine learning-based attack and anomaly detection system to protect against and lessen IoT cybersecurity risks in smart cities. Ensemble methods like bagging, boosting, stacking as an alternative

to earlier attempts that have concentrated on single classifiers to enhance the detection scheme. The incorporation of feature selection, cross-validation, and multiple class categorizations were considered. It provides greater specificity and maximal encryption time.

Chohan et al. [20] have suggested an intrusion detection scheme for IoT-based smart cities that relies on machine learning to find cyber-attacks. The UNSW-NB15 dataset was used. ADABoost, Multi-Layer Perceptron, Auto Encoder Classifier, Linear Support Vector Machine, and Quadratic Support Vector Machine were evaluated. It provides lesser encryption time with lower accuracy.

Prabakar et al. [21] have suggested Energy Analysis-Based Cyber Attack Identification by IoT with Artificial Intelligence at the Sustainable Smart City. A traffic analysis was conducted using kernel quadratic vector discriminant mechanism that improves data transmission through lowering network traffic. As a result of less traffic, energy efficiency was improved. The adversarial Bayesian belief networks were used to detect the malicious attacks. It provides lower computational time with low specificity.

While there are existing methods for detecting attacks and securing data transmission in IoT networks, many of these methods have limitations. Traditional approaches often struggle with higher false positive rates, lack adaptability to evolving attack patterns, and fail to provide robust security in the dynamic and heterogeneous environment of smart cities. This creates a pressing need for innovative solutions that effectively identify and mitigate attacks while ensuring secure data transmission. Table 1 represents the comparison of related work.

## 3 Proposed methodology

In this manuscript, cycle-consistent generative adversarial network for Attack detection on internet of things based SC and secures data transmission utilizing optimized advanced encryption (IoT-SC-CCGAN-ADSDT) is proposed. The IoT-SC-CCGAN-ADSDT approach is illustrated in Fig. 1. The detailed description of IoT-SC-CCGAN-ADSDT method is given below,

### 3.1 Data acquisition

Input data is obtained using NSL-KDD data set [22]. NSL-KDD is a widely used benchmark dataset for evaluating network IDS. It consists of a comprehensive set of network traffic records, with 125,973 training instances and 22,544 testing instances, each characterized by 41 features. These features include both continuous and categorical attributes related to network connections and activities. The dataset categorizes instances into normal connections and various attack types (normal and attack), providing a valuable resource for developing and testing IDS. It is designed to address some of the limitations of original KDD Cup 1999 dataset, such as redundancy and class imbalance. Table 2 represents the NSL-KDD dataset features.

### 3.2 Pre-processing

Here, input data is pre-processed. Then, the pre-processing comprises three steps: crisp data conversion, splitting, normalization. The detailed discussion regarding these steps as follows; initially, input data is transmitted into crisp data conversion. The crisp data contains certain string value data's, like Transmission Control Protocol, Internet Control Message Protocol. These string values are not processed, but string values are converted to numbers. For example, Transmission Control Protocol data is indicated as one and Internet Control Message Protocol data are specified as two. Other data specified as numbers, this is known as crisp data conversion. After the conversion, crisp data is splitted as attack type, like DoS, remote to local attacks, user-to-root, probe attack, normal. After splitting the data, split is regulated with Min–Max normalization. It is used to validate the information between 0 and 1 expressed in Eq. (1),

$$N_r = \left( \left( \frac{(J - J_{\min})}{(J_{\max} - J_{\min})} \right) \times (1 - 0) + 0 \right) \tag{1}$$

from Eq. (1), $N_r$ specifies normalized outcome, input data is represented by $J$, $J_{\min}$ denotes minimum value of data, $J_{\max}$ denotes maximal value of data.

### 3.3 Feature selection using wild horse optimizer (WHO)

In pre-processing step, each uncertainty is removed from the dataset; it selects the optimal features by using Wild horse optimizer (WHO) [23]. The main aim of using feature selection process lessens over fitting and training period, but raises the accuracy detection. The optimum feature selection reduces the time of feature learning. The proposed WHO algorithm solves problems with present optimization, nonlinear programming, and single goal optimization. Also, it is applied for solving the unconstrained and constrained optimization problems. The stepwise procedure for wild horse optimizer method based on feature selection procedure is specified below,

Step 1: Initialization

The populace of wild herd horse is initialized. Hence, the population of wild herd horse is expressed in Eq. (2),

**Table 1** Comparison of related work

| Author name | Title | Architecture | Characteristics | Advantages | Disadvantages | Research gap | Practical Applications |
|---|---|---|---|---|---|---|---|
| Alhanaya and Al-Shqeerat et al.,2023 | Creating an Incorporated approach for Machine Learning-Based IoT Traffic Security | Principal Component Analysis, ensemble classifier | To deal safe data communication in smart cities | It attains lesser computational time | It attains lesser accuracy | Lack of integrated machine learning frameworks for both attack detection and secure data transmission in IoT traffic | Machine learning for traffic management, but it doesn't specify practical applications |
| Rashid et al.,2022 | Applications based on IoT that use adversarial training to identify cyberattacks in smart cities | Machine learning | To upgrade IDS performance when confronting adversarial attacks | It provides higher accuracy | It provides lower sensitivity | Research lacks exploration of adversarial training effectiveness in diverse IoT applications | To detect attacks on IoT devices in smart cities and improve data security during transmission |
| Al-Farhani et al.,2023 | IoT and BC-Basally Cloud Model for Safe Data Transmission | GA-DBN algorithm | To transmit data securely over IoT, cloud-based blockchain security architecture has been suggested | It provides better specificity | It provides lower security level | Limited exploration of blockchain technology with cloud-based models for enhancing data | To develop real-time intrusion detection systems for smart city infrastructure |
| Rashid et al.,2020 | Cyberattacks detection in iot-dependsmart city applications with the help of machine learning models | SVM, ANN | For a smart city's IoT cybersecurity defense and mitigation | It provides lower encryption time | It provides lower precision rate | Cyberattack detection in IoT-based smart city applications | This approach could reduce the risk of cyberattacks that could disrupt traffic flow or endanger public safety |
| Chohan et.al., 2023 | Recognition of Cyber Attacks under Machine Learning basis IDS for IoT based Smart Cities | LSVM, QSVM, ADA Boost | Exploration of ML approaches trained over UNSW-NB15 data set | It offers higher specificity | It provides lower accuracy | Intrusion Detection Systems (IDS) specifically tailored for the unique challenges of IoT-based smart cities | This technology could be applied to secure data transmission between wearable health monitors |
| Prabakar et.al.,2023 | Cyber Attack Detection depends on Energy Analysis-through IoT along Artificial Intelligence in Smart City | Artificial intelligence | IoT artificial intelligence techniques for cybersecurity-based network traffic analysis and malevolent attack finding for a sustainable smart city | It provides lower computational time | It provides lower specificity | synergistic potential between energy analysis via IoT and artificial intelligence for cyber-attack detection | could safeguard financial transactions within smart cities by securing data transmission |

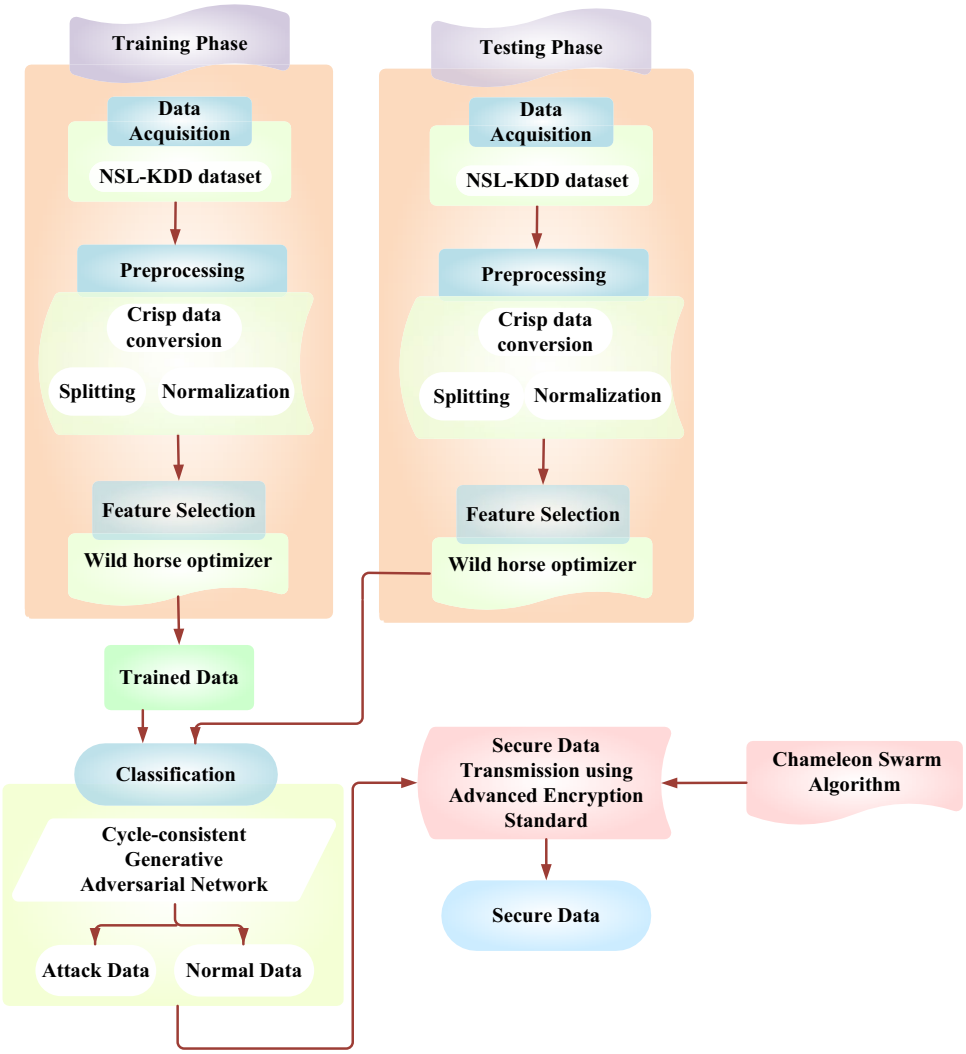


**Fig. 1** Proposed IoT-SC-CCGAN-ADSDT method

$$population\ herd = \{population\ herd_1, population\ herd_2, .....population\ herd_n\} \quad (2)$$

where $n$ denotes the herd of $n$ horses.

Step 2: Random Generation

Create the input parameters at random after initialization. Optimal fitness values of each Wild herd horse are the final results of selected features based upon clear hyper parameter circumstances.

Step 3: Fitness Function

It is utilised for increasing the accuracy, less computational time as well as obtain the objective function. Hence, the fitness value of horses from herd function is expressed in Eq. (3),

$$Fitness_{function} = \frac{population(herd_p)}{n} \quad (3)$$

where $herd_p$ denotes the rank of each horse, such that $p \in \{1, .....n\}$.

Step 4: Dynamics of Stallion

In this step, the center of the herd function is generally used for obtaining the objective function formulated in Eq. (4),

$$f_{RF}(H) = \frac{\sum_{p=1}^{n}(z_p \times herd_p.rank)}{\sum_{p=1}^{n} herd_p.rank} \quad (4)$$

where $z_p$ denotes the position of the horse from the herd.

Step 5: Grazing Behavior of Horses

It describes foal grazing among its family. By the grazing behaviour of horses, velocity updating is given by Eq. (5),

$$V_{p.q}^{T+1} = V_{p.q}^{T} + R \times herd_p.gait \times \left(Nherd_{centerq}^{T} - z_{p.q}^{T}\right) \quad (5)$$

**Table 2** NSL-KDD dataset features

| Sl. No | Features | Category |
|---|---|---|
| 1 | protocol_type | Symbolic |
| 2 | land | Symbolic |
| 3 | flag | Symbolic |
| 4 | service | Symbolic |
| 5 | src_bytes | Continual |
| 6 | dst_bytes | Continual |
| 7 | num_shells | Continual |
| 8 | num_outbound_cmds | Continual |
| 9 | wrong_fragment | Continual |
| 10 | rerror_rate | Continual |
| 11 | urgent | Continue |
| 12 | dst_host_amount | Continue |
| 13 | is_host_login | Continue |
| 14 | dst_host_srv_diff_host_rated | Continue |
| 15 | dst_host_similar_src_port_rated | Continue |
| 16 | dst_host_same_srv_rated | Continue |
| 17 | dst_host_serror_rated | Continue |
| 18 | dst_host_srv_rerror_rated | Continue |
| 19 | dst_host_srv_serror_rated | Continue |
| 20 | dst_host_rerror_rate | Continual |
| 21 | hot | Continue |
| 22 | srv_diff_host_rate | Continue |
| 23 | count | Continue |
| 24 | dst_host_diff_srv_rate | Continue |
| 25 | srv_serror_rate | Continue |
| 26 | srv_rerror_rate | Continue |
| 27 | same_srv_rate | Continue |
| 28 | srv_count | Continue |
| 29 | num_compromised | Continue |
| 30 | num_access_files | Continue |
| 31 | num_file_creations | Continue |
| 32 | serror_rate | Continue |
| 33 | duration | Continue |
| 34 | num_failed_logins | Continue |
| 35 | su_attempted | Continue |
| 36 | diff_srv_rate | Continual |
| 37 | num_root | Continue |
| 38 | root_shell | Continue |
| 39 | is_host_login | Symbolic |
| 40 | is_guest_login | Symbolic |
| 41 | logged_in | Symbolic |

where $V_{p,q}^{T+1}$ denotes the updated velocity of horse herd, $R$ is a random number, $herd_p.gait$ specifies random number from the interval [−2,2], and $Nherd$ specifies nearest herd.

Step 6: Horse Mating Behavior for the selection of features

**Table 3** Features selected under WHO

| Sl. No | Features | Category |
|---|---|---|
| 1 | protocol_type | Symbolic |
| 2 | src_bytes | Continual |
| 3 | num_outbound_cmds | Continual |
| 4 | wrong_fragment | Continual |
| 5 | urgent | Continual |
| 6 | dst_host_count | Continue |
| 7 | dst_host_same_srv_rate | Continual |
| 8 | dst_host_srv_rerror_rate | Continual |
| 9 | dst_host_rerror_rate | Continual |
| 10 | srv_diff_host_rate | Continual |
| 11 | dst_host_diff_srv_rate | Continual |
| 12 | srv_rerror_rate | Continual |
| 13 | srv_count | Continual |
| 14 | duration | Continue |
| 15 | diff_srv_rate | Continue |
| 16 | num_root | Continue |
| 17 | root_shell | Continual |
| 18 | logged_in | Symbolic |
| 19 | is_guest_login | Symbolic |
| 20 | num_file_creations | Continual |

It specifies the behaviour of horses when they leave their families and joins with single groups. Hence, the position of horses is represented using Eq. (6),

$$F_s\left(H_t, u\right) = z_{p,q}^{T} + V_{p,q}^{T+1} \tag{6}$$

where $T$ specifies current iteration, $T+1$ specifies new iteration, $p \in \{1, .....P\}$ specifies index of features selected and $q \in \{1, .....L\}$ specifies the index features of dimension.

Step 7: Selecting and Exchanging Leaders

This phase describes the selection of features. Initially, the features are selected randomly, and expressed in Eq. (7),

$$M_{n,p,q}^{T+1} = z_{p,q}^{T+1} \times W(0, sd) \tag{7}$$

here $M_{n,p,q}^{T+1}$ represents the optimal selected features, $n \in \{1, .....HMP\}$, $HMP$ means Horse memory pool value, $W(0,1)$ specifies normal distribution along mean zero and standard deviation $sd$, $z_{p,q}^{T+1}$ denotes the dimension of the features.

Step 8: Termination

End the process afterwards achieving optimum solution from first solution. When attains optimum solution, step 3

will repeat till halting criteria met. Finally, WHO selects 20 ideal features for maximizing the classification speeds and minimizing the computation time. Table 3 displays selected 20 features.

### 3.4 Classification using CCGAN

The extracted features are given to CCGAN [24] for categorization that categorizes the data as attacked or non-attacked. Normally, CCGAN trains generator $G$ and discriminator $D$. $G$ Generator captures data discernment for producing newly samples. The generated samples novelty is found out through $D$ discriminator. CCGAN is differing from original GAN. Also, it has two $G_{xy}$ *and* $G_{yx}$ generators as well as two $D_x$ *and* $D_y$ discriminators. The max–min complexity is determined utilizing Eq. (8),

$$\min_{G_{xy},G_{yx}D_x,D_y} \max L_{GAN}(G_{xy}, D_y) + L_{GAN}(G_{yx}, D_x) \tag{8}$$

where

$$\begin{aligned} L_{GAN}(G_{xy,yx}, D_{x,y}, X, Y) = \mathrm{E}_{a,b\approx PX,PY}\left[\log D_{x,y}(a,b)\right] \\ +\mathrm{E}_{a,b\approx PX,PY}\left[1-\log(D_{x,y}(G_{xy,yx}(a,b)))\right] \end{aligned} \tag{9}$$

where $(G_{xy}, D_y)$ and E signifies observed value including predicted operator. $G_{yx}(b)$ specifies generator outcome and $X, Y$ implies identity loss; $x$, y, $z$ signifies custom hyper parameters, $D_x(a)$; $D_x(G_{yx}(b)$ signifies discriminator output of actual samples $x$, generated samples $G_{yx}(b)$ refers input to the discriminator. Generator attempts for maximizing as well as minimizing the function $L_{GAN}(G_{yx}, D_x)$. For the created attack categories as sensible, joint loss operation is deemed and comprises cycle consistency, adversarial, and identity losses. The min–max complexity is considered as minimization complexity expressed in Eq. (10)

$$\min_D v(D) = \frac{1}{2}\mathrm{E}_{\delta\approx P_\delta}\left[(D(\delta)-y^2)\right] + \frac{1}{2}\mathrm{E}_{\eta\approx P_\eta}\left[(D(G(\eta)-x^2)\right] \tag{10}$$

where E(.) refers expectations, source with destination data represents $\delta$; $\eta$, $x$ and $y$ refers customs hyper parameter, $P_\eta$ denotes data sharing $x = 0$ and $y = 1$. It is not possible to generate output from discrete input for using conflict loss. To build that the generator generate attack identification from cloud computing that are related to better feature selection outcome, cycle consistency loss is applied $G_{xy}$ and $G_{yx}$, lessening the attack identification and their inputs $a$, $b$. Cycle Consistency Loss is calculated using expressed Eq. (11),.

$$\begin{aligned} L_{cycle}(G_{xy}, G_{yx}) = \mathrm{E}_{a\approx PX}\left(\|G_{yx}(G_{xy}(a)) - a\|_1\right) \\ +\mathrm{E}_{b\approx PY}\left(\|G_{xy}(G_{yx}(b)) - b\|_1\right) \end{aligned} \tag{11}$$

where $\|\bullet\|_1$ specifies $L1$ norm. Cycle constancy with identity loss is used at combination. Identity loss determines better feature selection output. To diminish $L1$ length among $G_{yx}(a)$ and $a$ or among $G_{xy}(b)$ identity loss is considered and exhibited in Eq. (12),

$$\begin{aligned} L_{identityloss}(G_{xy}, G_{yx}) = \mathrm{E}_{a\approx PX}\left(\|G_{yx}(a) - a\|_1\right) \\ +\mathrm{E}_{b\approx PY}\left(\|G_{xy}(b)) - b\|_1\right) \end{aligned} \tag{12}$$

Total loss operation is labeled in Eq. (13),

$$\begin{aligned} L_{Total\ Loss} = L_{GAN}(G_{xy}, D_y, X, Y) + L_{GAN}(G_{yx}, D_x, X, Y) \\ +\gamma_1 L_{cycle}(G_{xy}, G_{yx}) + \gamma_2 L_{identity\ loss}(G_{xy}, G_{yx}) \end{aligned} \tag{13}$$

Generators $G_{xy}$ and $G_{yx}$ is designed. From optimal selection of feature output, features extracted. Here, 12 convolutional layers set is organized via univariate ensemble base feature selection. A 2 dimensional convolution kernel by means of $1\times1$, $3\times3$, and $5\times5$ spatial sizes is used to remove features of dissimilar scales on the basis of BN activation and leaky corrected linear unit represented Eq. (14),

$$f(z) = \begin{cases} z, if & z > 0 \\ \mu z, if & z \le 0 \end{cases} \tag{14}$$

Consider $\mu$ as controlling parameter and its value is 0.25. CCGAN discriminator module distinguishes among secure data and secrecy attack data categories of IoT. The $\wp_{dis}$ parameter in discriminative module left stable and $\wp_{gen}$ parameter in creator component enhanced through Eq. (15),

$$\wp_{gen} \leftarrow \wp_{gen} + \mathfrak{I}\frac{\partial Obj_{gen}}{\partial \wp_{gen}} \tag{15}$$

here $\mathfrak{I}$ implies learning rate, $Obj_{gen}$ implies attack detection assessed by $\xi_{smb}$ choosing example event at random, $\xi_{smb}$ denotes mini-batch size. CCGAN not only acts well in training data, but also avert over-fitting issue. Every ideal features selected under Wild horse optimizer algorithm base feature selection learns in each CCGAN layer, and then better level features are categorized. Parameter. $\wp_{dis}$ in discriminator component examines with parameter $\wp_{gen}$ in generator component remains stable is articulated in Eq. (16),

$$\wp_{dis} \leftarrow \wp_{dis} - \mathfrak{I}\left(\frac{\partial Obj_{gen}}{\partial \wp_{dis}} - \frac{\partial Obj_{dis}}{\partial \wp_{dis}}\right) \tag{16}$$

here $Obj_{dis}$ denotes secrecy attack recognition determined by $\xi_{smb}$ selected randomly from the benchmark data set of NSL-KDD. Finally, the proposed CCGAN classified the data as attack and non-attack. Attacked and non-attacked (normal) are the two types of data that are available at the output of categorization phase. The attack type precisely identifies the attack data. It retains in log file. The cloud- server manages for applications. The exploration of normal data is

carried out for prediction of various issues, such as traffic manage, structural health monitor, pollution prevent, parking optimization, waste management, intelligent transport, smart buildings.

### 3.5 Advanced encryption standard (AES) for secure data transmission

After revealing the outcome using Advanced Encryption Standard (AES) and its output is sent to the user. This AES cryptographic approach provides secure layer to secure the data and lessens the time complexities. The advanced encryption algorithm is symmetric encryption proves it's effect at more application. The basic algebraic structure including same replication method encrypts all the blocks in the signal, such as loopholes that makes the signal vulnerable to different kinds of attacks. Thus, the encryption and decryption techniques are implemented using generated keys. Therefore, the binary security decision for user $i$ is expressed in Eq. (17),

$$\xi_i \in \{0,1\} \tag{17}$$

here $\xi_i = 0$ denotes unsecured data for the user $i$ computation task is offloaded and $\xi_i = 1$ specifies secure data for user $i$ computation task is encrypted by utilizing security layer. The data privacy through the computation task depends on user behavior is expressed in Eqs. (18) and (19),

$$E_{i,j}^{ep} = u_i ep_{i,j} \tag{18}$$

$$T_{i,j}^{ep+dp} = \frac{ep_{i,j}}{g_i^l} + \frac{dp_{i,j}}{g_i^e} \tag{19}$$

where $E_{i,j}^{ep}$ denotes the energy overhead and $T_{i,j}^{ep+dp}$ denotes the additional time, $ep_{i,j}$ and $dp_{i,j}$ represents the central processing unit cycles to encrypt and decrypt the computation task's data at user $i$ including edge server. The total overhead is needed for transmitting the computational task $j$ remotely to user $i$ at small base station after applying the security decision that is calculated by expressed Eq. (20),

$$T_{i,j}^{\text{Security}} = \left[ T_i \left( s_{i,j}^{ep+dp} + T_{i,j}^{trns} \right) + \left(1 - \xi_i\right) T_{i,j}^{trns} \right] \tag{20}$$

where $T_{i,j}^{\text{Security}}$ represents the security decisions, $T_{i,j}^{trns}$ denotes the transmission of data and $s_{i,j}^{ep+dp}$ represents the data encryption and decryption. The executing tasks of user security models are computed by expressed Eq. (21),

$$T_{i,j} = \left[ \phi_{i,j,0} T_{i,j}^{l} + \phi_{i,j,n+1} (T_{i,j}^{\text{security}} + \vartheta + T_{i,j}^{EExe}) + \sum_{n=1}^{N} \phi_{i,j,\gamma} (T_{i,j}^{\text{security}} + T_{i,j}^{EExe}) \right] \tag{21}$$

where $\phi_{i,j,0} T_{i,j}^{l}$ represents the local time execution, $\phi_{i,j,n+1}(T_{i,j}^{\text{security}} + \vartheta + T_{i,j}^{EExe})$ and $\sum_{n=1}^{N} \phi_{i,j,\gamma}(T_{i,j}^{\text{security}} + T_{i,j}^{EExe})$ represents cloud and edge time implementation for protection layer. By protecting the global execution, computational tasks are safely implemented locally. If not, encrypts the computation task and it is uploaded to the cloud server for remote execution. Therefore, the systems weighted cost to demands energy as well as time implements the task $j$ of user $i$ is expressed in Eq. (22),

$$W_{i,j} = z_i^s T_{i,j} + z_i^e E_{i,j} \tag{22}$$

here $z_i^s T_{i,j}$ and $z_i^e E_{i,j}$ represents the weight parameters of computation energy and time to user $i$,$z_i^s T_{i,j}$ *and* $z_i^e E_{i,j} \in [0,1]$. The attained outcomes indicate that the proposed advanced encryption standard (AES) technique has efficient data transmission. To get more secure data transmission weight parameters $z_i^s T_{i,j}$ and $z_i^e E_{i,j}$ of the advanced encryption standard technique is optimized by CSA. The comprehensive description of Chameleon Swarm algorithm is described.

#### 3.5.1 Stepwise process of chameleon swarm algorithm for optimizing AES

CSA is used to enhance the parameters of the advanced encryption standard (AES) technique to obtain the ideal parameters. Hence, these parameters are enhanced by scaling the ideal parameters to secure data transmission. Since chameleons are generally distinct and highly specialized among a wider range of species, their ability to blend in with their surroundings through color changes helps to identify them. Chameleons hunt in the desert and among trees, much like all other organisms in nature, therefore their location fluctuates. The safe data transmission is achieved by using chameleon behaviors such as tracking and searching, as well as catching and hunting prey. The stepwise process of Chameleon Swarm Algorithm (CSA) [24–26] shown beneath,

Step 1: Initialization

The chameleons population $k$ through $d$-dimensional search space are initiated with 2D matrix $x$ depending on the dimension $k \times d$. It is expressed in Eq. (23),

$$x_i^n = \left( x_{i,1}^n, x_{i,2}^n, \ldots\ldots\ldots x_{i,d}^n \right) \tag{23}$$

where, $x_i^n$ implies position of $n^{th}$ data at $d^{th}$ dimensional space.

Step 2: Random generation

Here, the uniform random value is created using population initialization. The position of each data is computed randomly and expressed in Eq. (24),

$$x^n = q_m + R(p_m - q_m) \tag{24}$$

Here $p_m, q_m$ implies bounds of secure area using $m^{th}$ dimension, $R$ signifies uniformly generated random number in [0,1] range.

Step 3: Fitness Function

Fitness functions are examined to obtain objective function, like secure data transmission and for attaining optimal value. Here, the weight parameters of advanced encryption standar d $= z_i^s T_{i,j}$ and $z_i^e E_{i,j}$ are enhanced utilizing the chameleon swarm algorithm (CSA). CSA is articulated in Eq. (25),

$$Fitness = optimization[z_i^s T_{i,j} \; and \; z_i^e E_{i,j}] \tag{25}$$

Step 3.a*:* Searching prey behaviour for optimizing weight parameter $z_i^s T_{i,j}$

Chameleon's displacement behaviour to search prey is used for optimizing the weight parameter $z_i^s T_{i,j}$. Then, objective function is transmission of data with more security and expressed in Eq. (26),

$$K = Wc_1(r_E/i_E) + Wc_2(A_q/I_q) + Wc_3(R_{mn}) + Wc_4(d/c) + Wc_5(a_B/r_B) \tag{26}$$

where $Wc = Wc_1, Wc_2, .....Wc_5 = 1$ denotes the weighted co-efficient and $r_E, i_E$ implies the energy values, $A_q$, $I_q$ signifies initial data size and current accessible data. The reliability of secure path $m, n$ is denoted as $R_{mn}.a_B$, $r_B$, $d$, and $c$. implies required bandwidth, residual bandwidth, distance, and coverage.

Step 4: Hunting prey behaviour for optimizing weight parameter $z_i^e E_{i,j}$

Normally, when the target gets closer, chameleons employ their hunting behavior to attack it with their tongue. This updates the chameleons' positions and brings them near to prey. This behaviour is employed to optimize the weight parameter $z_i^e E_{i,j}$. Thus, data prioritization by position updation of chameleons is expressed in Eq. (27),

$$V_{i+1}^{m,n} = x_i^n \times WcV_i^{m,n} + C1\left(bP_i^{m,n} - A_i^{m,n}\right)r1 + C2\left(lc_{m,n}d^2 - A_i^{m,n}\right)r2.h_{m,n} \tag{27}$$

where $A_i^{m,n}$ denotes actual position of data, $bP_i^{m,n}$ implies the best secure path, $r1$, $r2$, denotes random number in [0,1] range $lc_{m,n}$ w signifies link cost function with $d$ distance, $h_{m,n}$ denotes number of hops controlled using positive integers $C1, C2$. Thus, the condition $x_i^n = bP_i^{m,n}$ is evaluated and data transferred in safe manner. Figure 2 portrays the flowchart of chameleon swarm algorithm.

Step 5: Termination Condition

Here, the optimal hyper-parameter $z_i^s T_{i,j}$ and $z_i^e E_{i,j}$ are chosen in advanced encryption standard (AES) technique with CSA, will repeating step 3 iteratively till halting criteria met**.** By this, AES transmit data more securely by using chameleon swarm algorithm.

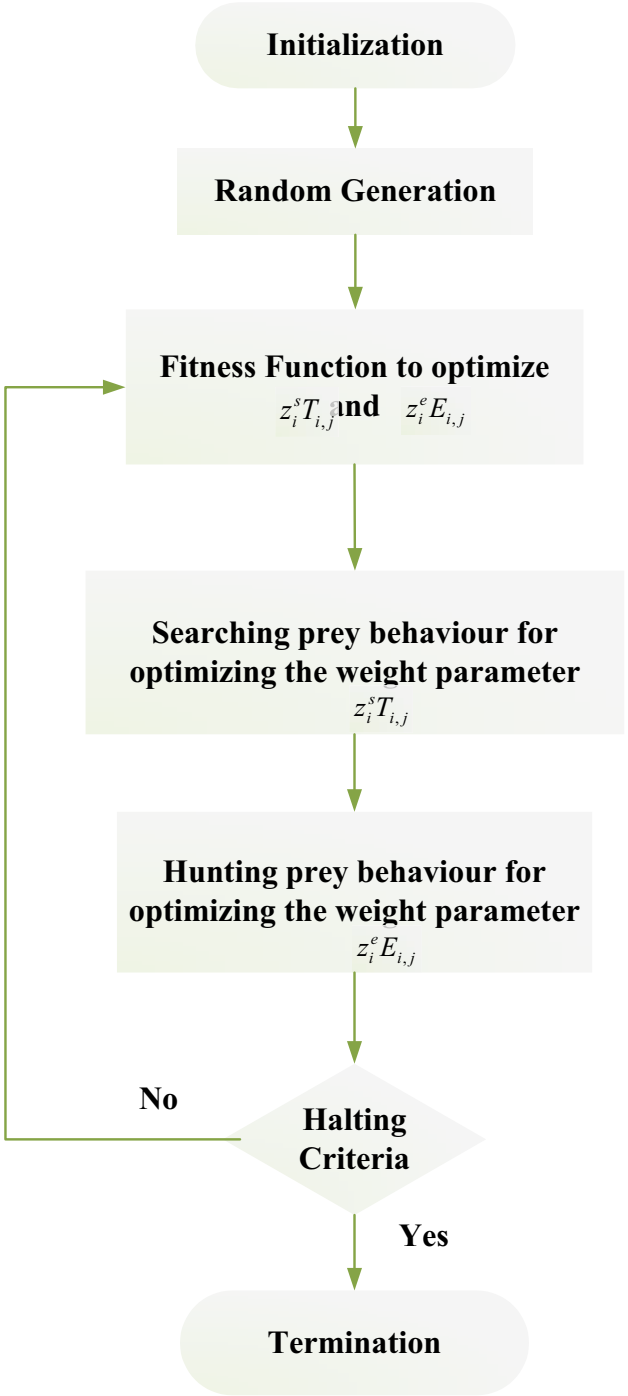


**Fig. 2** Chameleon swarm algorithm for optimizing AES

After transferring the information safely, the encrypted data decrypts in terms of chameleon swarm algorithm. Then display the decrypted normal data in LCD monitor.

## 4 Results and discussions

The simulation performance of CCGAN is proposed for Attack Detection on internet of things Based SC and Secure Data Transmission Using optimized Advanced Encryption method. The proposed technique is done in java language utilizing Windows operating scheme with Intel core i7 CPU processor, 16 GB random access memory. The outcomes are compared with existing Adversarial Training for Deep Learning-based cyber-attack identification in IoT-based Smart City Applications (DL-IOT-SCA) [15], an Integrated Framework for Protecting IoT Traffic in Smart Cities under Machine Learning methods to identify suspicious traffic and deal with secure data transmission in smart cities (IoT-SC-PCA) [16], Enhanced IDS using Deep Learning for IoT-Based Smart Cities Security (IoT-SCA-DL) [17].

### 4.1 Performance metrics

The efficiency of the proposed method is evaluated under mentioned metrics. The following confusion matrix is needed to calculate the performance metrics.

- True Positive ($T_P$): normal accurately identified as normal
- True Negative ($T_N$): attack accurately identified as attack
- False Positive ($F_P$): attack inaccurately identified as normal
- False Negative ($F_N$): normal inaccurately identified as attack

#### 4.1.1 Accuracy

This is the ratio of accurate prediction with total proceedings in dataset. It is measured using Eq. (28),

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{28}$$

#### 4.1.2 Precision

The capability of the classifier to compute usual data lacking condition is measured by Eq. (29),

$$\text{Pr}ecision = {}^{T_P}\Big/_{(T_P+F_P)} \tag{29}$$

#### 4.1.3 Sensitivity

This calculates the accurately predicted actual positive and computed using Eq. (30),

$$Sensitivity = {}^{T_P}\Big/_{F_N+F_P} \tag{30}$$

#### 4.1.4 Specificity

This is termed as true negative rate and scaled using Eq. (31),

$$Specificity = {}^{F_N}\Big/_{F_P+F_N} \tag{31}$$

### 4.2 Performance analysis

Figures 3, 4, 5, 6, 7, 8, and 9 portrays the simulation outcomes of cycle-consistent generative adversarial network for Attack Detection on internet of things based SC and Secure Data Transmission Using optimized Advanced Encryption method. The performance metrics are examined and compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL methods.

Figure 3 depicts accuracy analysis for both normal and attack data. For normal data, the proposed IoT-SC-CCGAN-ADSDT method achieves 12.33% higher accuracy compared to the DL-IOT-SCA method, 6.77% higher accuracy than the IoT-SC-PCA method, and 10.8% higher accuracy over the IoT-SCA-DL method. For attack data, the IoT-SC-CCGAN-ADSDT method provides a 0.9% increase in accuracy compared to the DL-IOT-SCA method, a 0.66% improvement over the IoT-SC-PCA method, and a 1.3% higher accuracy than the IoT-SCA-DL method.

Figure 4 illustrates precision analysis for both normal and attack data. For normal data, the proposed IoT-SC-CCGAN-ADSDT method achieves 25.95% greater precision compared to the DL-IOT-SCA model, 6.37% higher precision than the IoT-SC-PCA model, and 0.13% greater precision over the IoT-SCA-DL model. For attack data, the IoT-SC-CCGAN-ADSDT method provides 19.16% higher precision compared to the DL-IOT-SCA model, 20.46% greater precision than the IoT-SC-PCA model, and 16.77% higher precision over the IoT-SCA-DL model.

Figure 5 shows sensitivity analysis. The proposed IoT-SC-CCGAN-ADSDT method provides 33.24%, 32.88% greater sensitivity compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL. For attack data, the proposed IoT-SC-CCGAN-ADSDT method provides 12.34%, 21.71% higher sensitivity evaluated with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models.

Figure 6 signifies specificity analysis. Here, the specificity of proposed IoT-SC-CCGAN-ADSDT method is measured and performances are likened with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models. For normal data, the proposed IoT-SC-CCGAN-ADSDT method

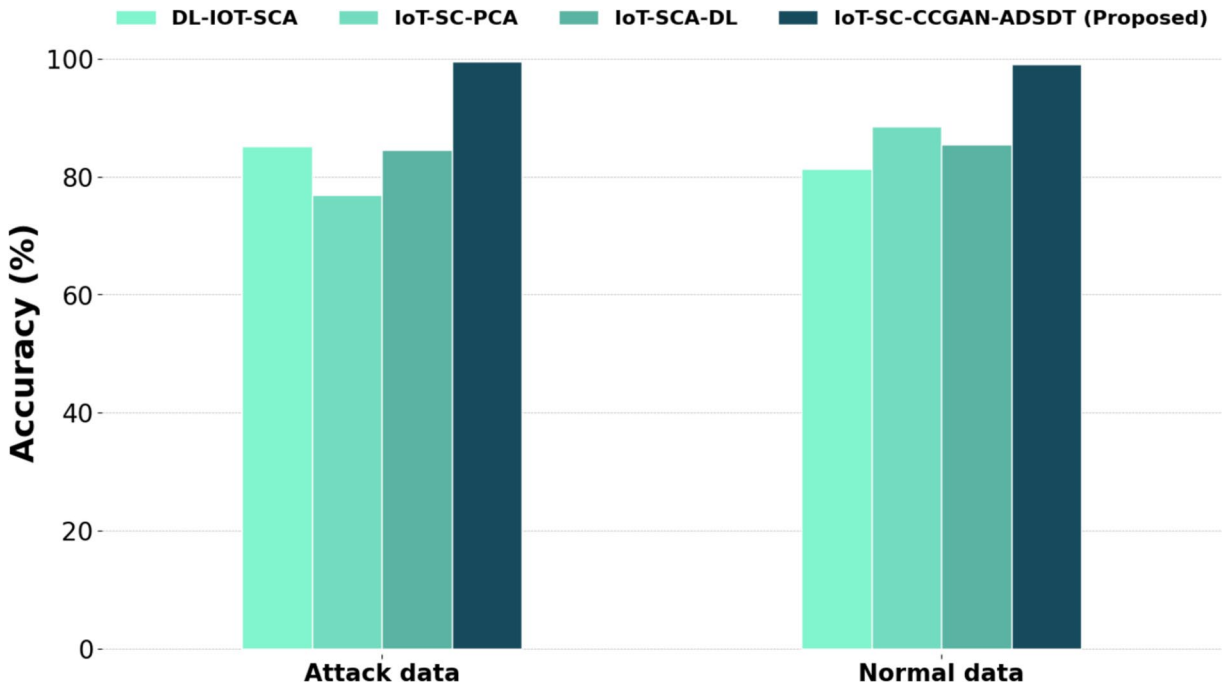


**Fig. 3** Accuracy analysis

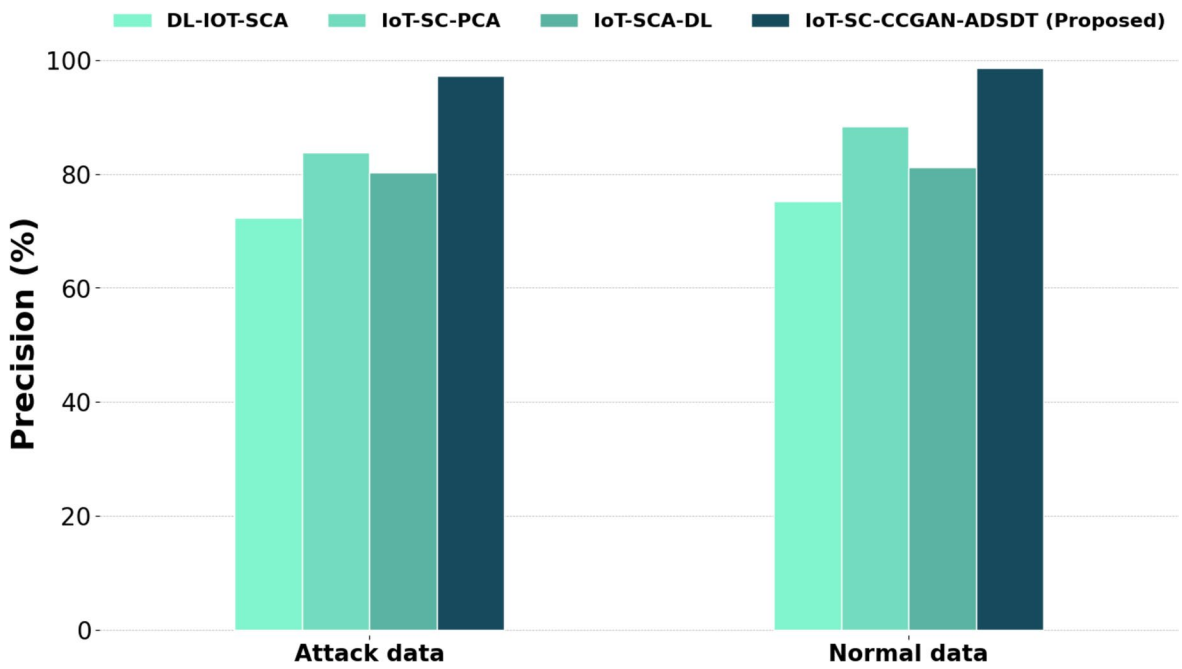


**Fig. 4** Precision analysis

provides 0.81%, 16.54% and 12.32% greater specificity evaluated with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models. For attack data, the proposed IoT-SC-CCGAN-ADSDT method provides 0.18%, 13.65% and 11.43% better specificity compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models.

Figure 7 implicates encryption time analysis. Here, encryption time of proposed IoT-SC-CCGAN-ADSDT method is measured and performances are compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models. For normal data, the proposed IoT-SC-CCGAN-ADSDT attains 18.01%, 7.7%,1.42% lesser encryption

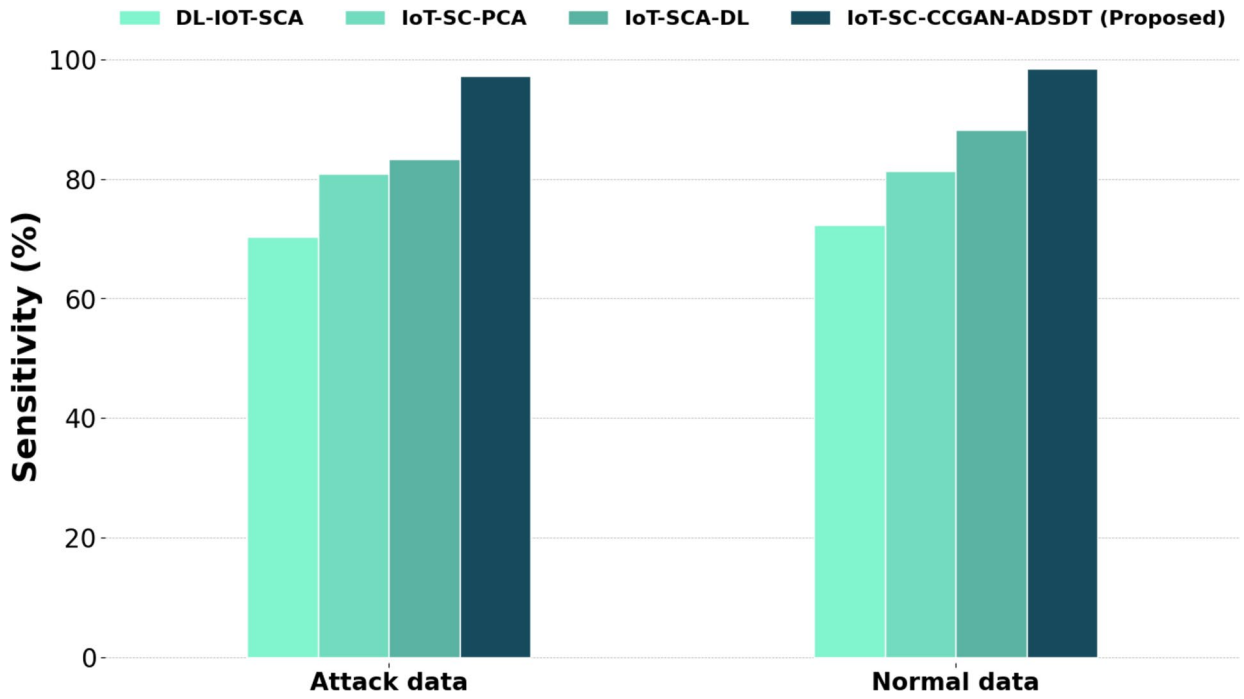


**Fig. 5** Sensitivity analysis

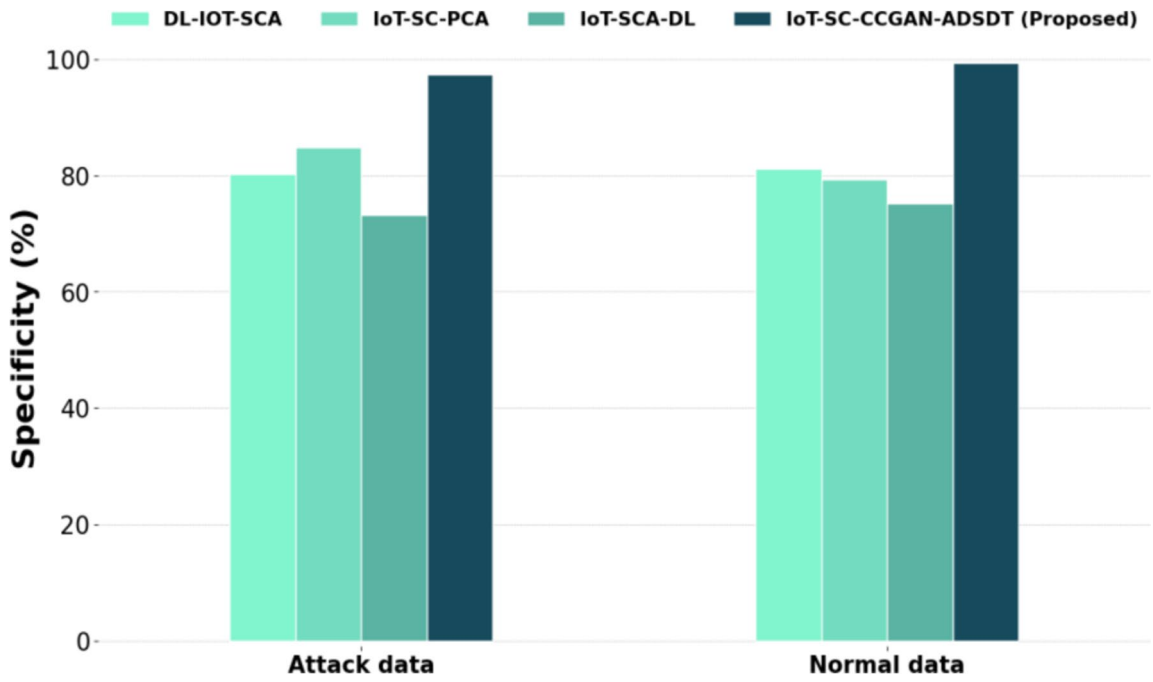


**Fig. 6** Specificity analysis

time compared to existing techniques. For attack data, the IoT-SC-CCGAN-ADSDT method provides 11.05%, 9.76% and 25.31% lesser encryption time compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models.

Figure 8 displays Decryption time analysis. Here, decryption of IoT-SC-CCGAN-ADSDT method is measured and performances are compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models. For normal

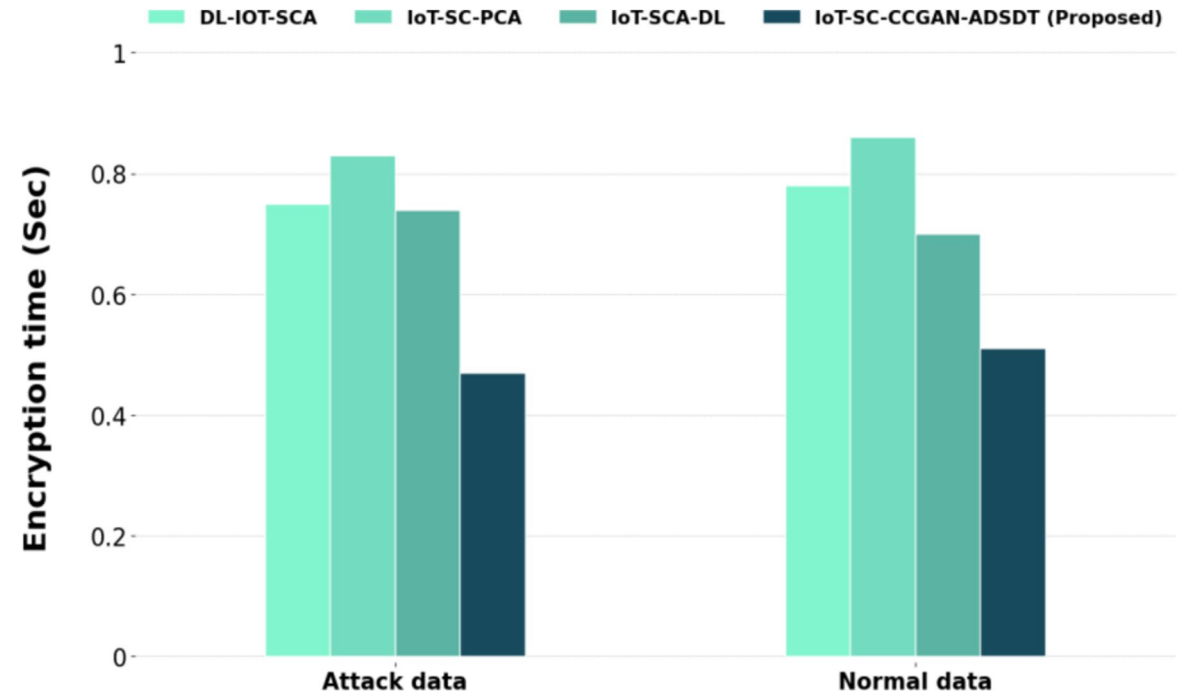


**Fig. 7** Encryption time analysis

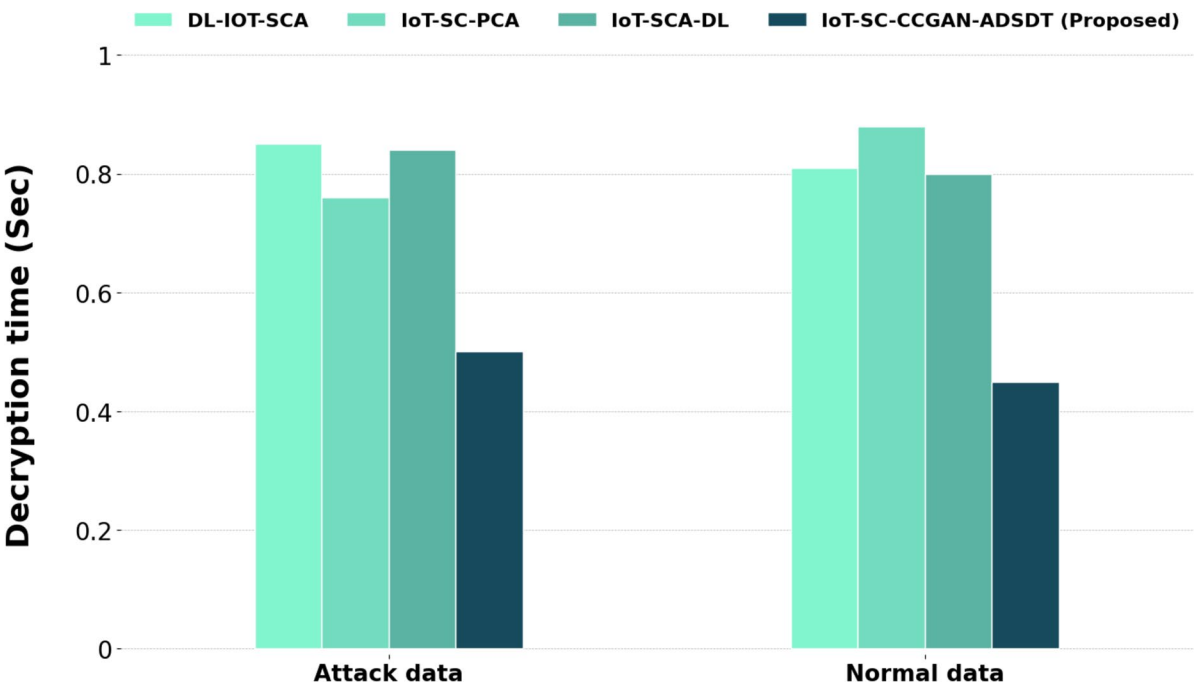


**Fig. 8** Comparative analysis of Decryption time

data, the proposed IoT-SC-CCGAN-ADSDT attains 22.06%, 10.39%, 12.23% lesser decryption time compared with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL. For attack data, the proposed IoT-SC-CCGAN-ADSDT method provides 32.06%, 25.97% and 34.12% lesser decryption time analyzed with existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL models.

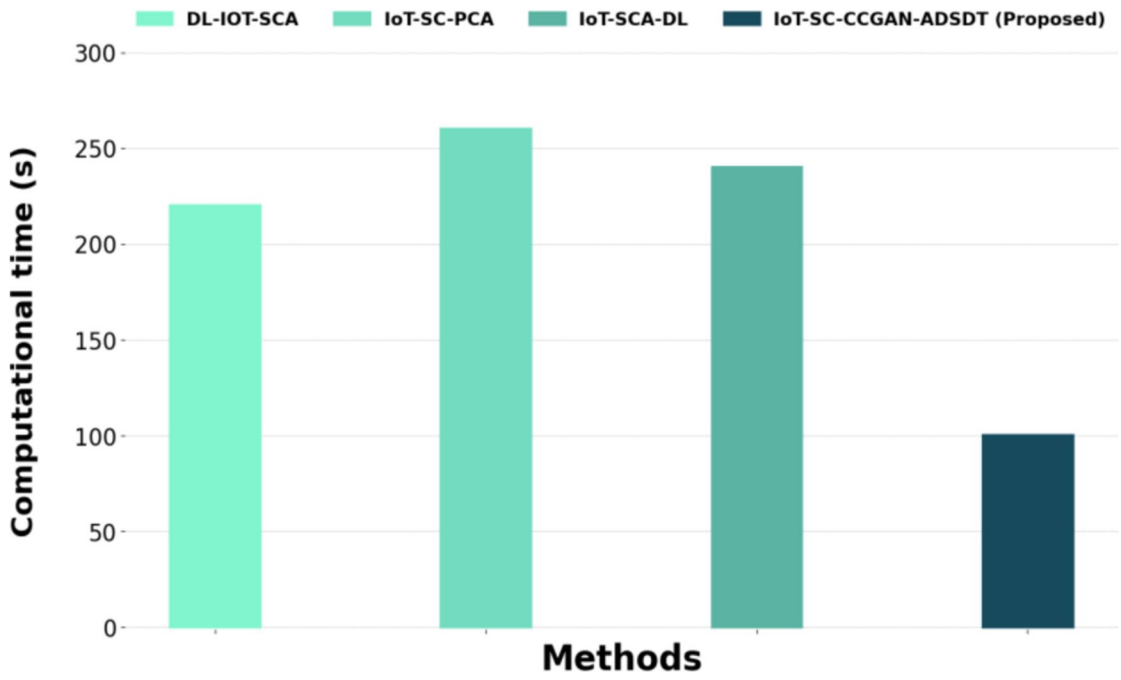


**Fig. 9** Computational time analysis

Figure 9 depicts the computational time analysis of IoT-SC-CCGAN-ADSDT method. The analysis shows that the IoT-SC-CCGAN-ADSDT method significantly reduces computational time, achieving 53% lower computational time compared to the DL-IOT-SCA model. It also provides a 12.58% reduction in computational time compared to the IoT-SC-PCA model and a 26.44% decrease compared to the IoT-SCA-DL model. These reductions demonstrate the efficacy of the proposed method in processing data more rapidly and making it a highly effective solution for real-time applications in IoT-based smart cities.

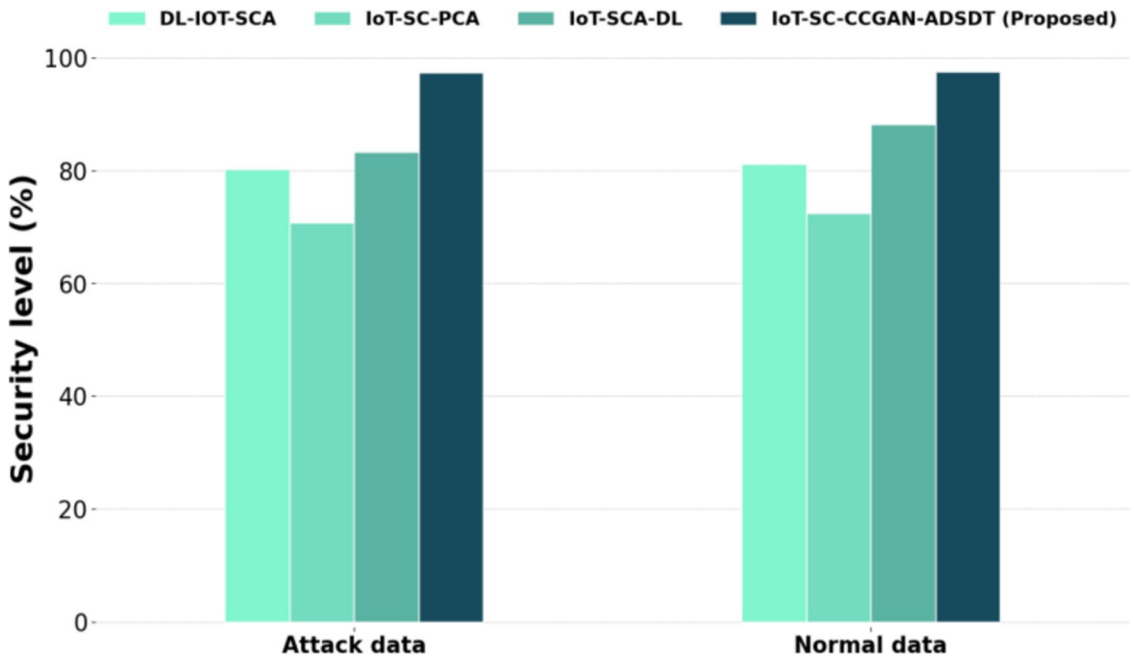


**Fig. 10** Security level analysis

Figure 10 illustrates security level analysis of the IoT-SC-CCGAN-ADSDT method compared to existing models. The IoT-SC-CCGAN-ADSDT method achieves a 21.08% higher security level than the DL-IOT-SCA model and a 6.07% improvement over the IoT-SC-PCA model. It also outperforms the IoT-SCA-DL model by 14.36% and another existing model by 3.67%. These results highlight the proposed method's superior capability in enhancing security for IoT-based smart cities.

## 5 Conclusion

In this manuscript, IoT-SC-CCGAN-ADSDT is implemented successfully. The propose method is evaluated using the NSL-KDD dataset. The IoT-SC-CCGAN-ADSDT method is examined using mentioned metrics. The IoT-SC-CCGAN-ADSDT method attains lower decryption time of 37.64%, 25.94% and 24.64 and lower encryption time 28.02%, 20.84%and 30.12% are compared with the existing DL-IOT-SCA, IoT-SC-PCA, IoT-SCA-DL methods. This framework combining Cycle-Consistent GANs with optimized AES encryption for enhanced attack detection and secure data transmission. Strengths include its novel approach, robust security measures, and comprehensive evaluation with the NSL-KDD dataset. However, it faces limitations such as dataset representation, scalability concerns, complexity of optimization, and the need for further validation across diverse IoT applications. Future research include the creation of appropriate attack models, the establishment of protocols for IDS performance evaluation, and the integration of IDS with other elements of the ICS security system, especially in light of the transition to the Future Internet environment.

**Acknowledgements** Not applicable

**Author Contribution** Mr. D.R. Denslin Brabin -(Corresponding Author)—Conceptualization Methodology, Original draft preparation Mrs. Agitha W -Supervision Mr.K. Kalai Kumar -Supervision Mrs.T. Sunitha -Supervision.

**Funding** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Mr. D.R. Denslin Brabin: Conceptualization, Methodology, Writing- Original draft preparation.

Mrs. Agitha W: Supervision.

Mr. K. Kalai Kumar: Supervision.

T. Sunitha: Supervision.

**Data availability** Data sharing does not apply to this article as no new data has been created or analyzed in this study.

## References

1. Mukherjee S, Gupta S, Rawlley O, Jain S (2022) Leveraging big data analytics in 5G-enabled IoT and industrial IoT for the development of sustainable smart cities. Trans Emerg Telecommun Technol 33(12):e4618
2. Huong TT, Bac TP, Long DM, Thang BD, Binh NT, Luong TD, Phuc TK (2021) Lockedge: low-complexity cyberattack detection in iot edge computing. IEEE Access 9:29696–29710
3. Ghosh S, Chandra V, Adhya A (2022) Machine Learning for Fog Computing-Based IoT Networks in Smart City Environment. In Intelligent Internet of Things for Healthcare and Industry. Springer, Cham, pp 267–285
4. Kumar P, Kumar R, Srivastava G, Gupta GP, Tripathi R, Gadekallu TR, Xiong NN (2021) PPSF: a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. IEEE Trans Netw Sci Eng 8(3):2326–2341
5. Inayat U, Zia MF, Mahmood S, Khalid HM, Benbouzid M (2022) Learning-based methods for cyber attacks detection in IoT systems: a survey on methods, analysis, and future prospects. Electronics 11(9):1502
6. Khoa TV, Saputra YM, Hoang DT, Trung NL, Nguyen D, Ha NV, Dutkiewicz E (2020) Collaborative learning model for cyberattack detection systems in iot industry 4.0. In: 2020 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp 1–6
7. Shajin FH, Rajesh P, Raja MR (2022) An efficient VLSI architecture for fast motion estimation exploiting zero motion prejudgment technique and a new quadrant-based search algorithm in HEVC. Circuits Systems Signal Process 41(3):1751–1774
8. Shajin FH, Rajesh P, Nagoji Rao VK (2022) Efficient framework for brain tumour classification using hierarchical deep learning neural network classifier. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, pp 1–8
9. Rajesh P, Shajin FH, Kumaran GK (2022) An Efficient IWOLRS Control Technique of Brushless DC Motor for Torque Ripple Minimization. Appl Sci Eng Prog 15(3):5514–5514
10. Rajesh P, Shajin FH, Kannayeram G (2022) A novel intelligent technique for energy management in smart home using internet of things. Appl Soft Comput 128:109442
11. Haque AB, Bhushan B, Dhiman G (2022) Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. Expert Syst 39(5):e12753
12. Alotaibi B, Alotaibi M (2020) A stacked deep learning approach for IoT cyberattack detection. J Sens 2020:1
13. Kantipudi MP, Aluvalu R, Velamuri S (2023) An Intelligent Approach of Intrusion Detection in Mobile Crowd Sourcing Systems in the Context of IoT Based SMART City. Smart Science 11(1):234–240
14. Ding W, Abdel-Basset M, Mohamed R (2023) DeepAK-IoT: an effective deep learning model for cyberattack detection in IoT networks. Inf Sci 634:157–171
15 Rashid MM, Kamruzzaman J, Hassan MM, Imam T, Wibowo S, Gordon S, Fortino G (2022) adversarial training for deep learning-based cyberattack detection in iot-based smart city applications. Comput Secur 120:102783
16. Alhanaya M, Al-Shqeerat K (2023) Developing an integrated framework for securing internet of things traffic in smart cities using machine learning techniques. Appl Sci 13(16):9476
17. Hazman C, Guezzaz A, Benkirane S, Azrour M (2024) Enhanced IDS with deep learning for IoT-based smart cities security. Tsinghua Science and Technology 29(4):929–947

18. Al-Farhani LH, Alqahtani Y, Alshehri HA, Martin RJ, Lalar S (2023) Jain R (2023) IOT and blockchain-based cloud model for secure data transmission for smart city. Secur Commu Netw 1:3171334
19 Rashid MM, Kamruzzaman J, Hassan MM, Imam T, Gordon S (2020) Cyberattacks detection in iot-based smart city applications using machine learning techniques. Int J Environ Res Pub Health 17(24):9347
20. Chohan MN, Haider U, Ayub MY, Shoukat H, Bhatia TK, Hassan MF (2023) Detection of cyber attacks using machine learning based intrusion detection system for IoT based smart cities. EAI Endorsed Trans Smart Cities 7(2):e4
21. Prabakar D, Sundarrajan M, Manikandan R, Jhanjhi NZ, Masud M, Alqhatani A (2023) Energy Analysis-Based Cyber Attack Detection by IoT with Artificial Intelligence in a Sustainable Smart City. Sustainability 15(7):6031
22. https://www.kaggle.com/datasets/hassan06/nslkdd
23. Naruei I, Keynia F (2022) Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. Eng Comput 38(Suppl 4):3025–3056. https://doi.org/10.1007/s00366-021-01438-z
24. Devi K, Muthusenthil B (2022) Intrusion detection framework for securing privacy attack in cloud computing environment using DCCGAN-RFOA. Trans Emerg Telecommun Technol 33(9):e4561
25. Braik MS (2021) Chameleon swarm algorithm: A bio-inspired optimizer for solving engineering design problems. Expert Syst Appl 174:114685
26. Preethi BC, Sugitha G, Kavitha G (2023) Cycle-consistent generative adversarial network optimized with water strider optimization algorithm fostered intrusion detection framework for securing cloud computing environment. Concurr Comput: Pract Exp 35(5):e7552




**AgithaW** received her B.Sc (CS) Degree from Malankara Catholic College, Mariagiri, India in 2002, M.Sc (CS) Degree from GVN College, Kovilpatti, India in 2004 and M. Tech in the specialization of Information Technology from Sathyabama University, Chennai in 2007. Also she has done Ph.D. at Bharath Institute of Higher Education and Research, Chennai, India. She has more than 15 years of teaching experience from various reputed Engineering colleges in Tamilnadu. Presently working as a faculty in the Department of Computer Science and Engineering, DMI College of Engineering, Chennai, India. Her research interests are Networks, Internet of Things, Machine Learning and Cloud Computing.

**D. R. Denslin Brabin** received B.E. and M.E. degrees in Computer Science and Engineering from Manonmaniam Sundaranar University, Tamil Nadu, India in 2002 and 2004 respectively. He received Ph.D. degree in Information and Communication Engineering from the Anna University, India in 2018. He has 18 years of teaching experience. Currently he is working as a Professor and Head in the Department of Computer Science and Engineering, DMI College of Engineering, Chennai, Tamil Nadu, India. He has published more than 20 research papers in International Journals. His current research interests include Cyber Security, Computer Networks and Image Processing

**K. Kalai Kumar** received the M.Tech. degree in Information Technology from Sathabama University, Chennai, Tamil Nadu, India, in 2008 and the Ph.D. degree in Information and Communication Engineering from Anna University, Chennai, Tamil Nadu, India, in 2020. Currently he is an Associate Professor, Department of Computer Science and Engineering, DMI College of Engineering College, Chennai, Tamil Nadu, India. His research interest includes Computer Networks and Network Security.

**T. Sunitha** received her B.E (CSE) Degree from Bhajarang Engineering College, Chennai, India in 2006 and M.E (CSE) from P. B. College of Engineering, Affiliated to Anna University, Chennai in 2013. She has more than 12 years of teaching experience from various reputed Engineering colleges in Tamilnadu. Presently working as Assistant professor and Head of Computer Science and Engineering department in P. B. College of Engineering, Chennai, India. Also she is currently a part time Ph.D. scholar at Anna University, Chennai, India. Her-research interests are Machine Learning, Deep Learning, Internet of Things (IoT), and Data science.