

Sailfish-cat algorithm-enhanced generative adversarial network for attack detection in internet of things-Fog network authentication

Pallavi Kanthamangala Niranjana¹, Ravikumar Venkatesh²

¹Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte, India

²Department of Information Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, India

Article Info

Article history:

Received May 13, 2024

Revised Aug 8, 2024

Accepted Sep 3, 2024

Keywords:

Attack detection

Fog computing

Generative adversarial networks

Internet of things

Optimization

ABSTRACT

The internet of things (IoT) has emerged as a prominent and influential concept within the realm of computing. Various attack detection methods are devised for detecting attacks in IoT-Fog environment. Despite all these efforts, attack detection still remained as a challenging task due to factors such as low latency, resource constraints of IoT devices, scalability issues, and distribution complexities. All these challenges are addressed in this paper by designing an efficient attack detection technique named as sailfish-cat optimization-based generative adversarial network (SaCO-based GAN) tailored for the IoT-Fog framework. This proposed approach introduces the SaCO-based GAN for IoT-Fog attack detection utilizing deep learning and feature-based classification, validated through experiments showing superior performance metrics. Notably, the SaCO optimization technique is utilized to train the GAN. Experimental results demonstrate the efficacy of the SaCO-based GAN with a maximum recall of 92.15%, a maximum precision of 91.21%, and a maximum F-Measure of 92.16%, outperforming existing techniques in IoT-Fog attack detection. The paper recommends enhancing scalability, implementing real-time detection strategies, rigorously testing robustness against diverse attack scenarios, and integrating with existing IoT security frameworks for practical deployment.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Pallavi Kanthamangala Niranjana

Department of Computer Science and Engineering, NMAM Institute of Technology, NITTE (Deemed to be University)

Karnataka, Nitte, 574110, India

Email: pallavi@nitte.edu.in

1. INTRODUCTION

The authenticated network of the internet of things Fog (IoT-Fog) environment is a complex ecosystem where IoT devices, Fog nodes, and cloud infrastructure collaborate to process and exchange data [1], [2]. This environment is characterized by its distributed architecture, diverse communication [3] protocols, and resource-constrained devices. Authentication mechanisms play a vital role in verifying the identity of entities within the network and ensuring secure data transmission as shown in Figure 1. Optimization algorithms aim to enhance the efficiency and effectiveness of various processes within this environment, including attack detection [4] strategies.

The IoT-Fog environment operates amidst significant scale and complexity, encompassing a vast array of interconnected devices, heterogeneous in their capabilities and communication protocols [5]. This diversity poses substantial challenges to scalability, as the network must efficiently accommodate the

expanding numbers of devices while maintaining robust performance and security measures [6]. Managing the heterogeneity of IoT devices adds another layer of complexity, necessitating adaptable authentication and detection strategies tailored to each device's unique characteristics. Moreover, the distribution of data across Fog nodes and cloud infrastructure introduces complexities in maintaining data integrity and confidentiality [7], demanding sophisticated security protocols. To mitigate risks, advanced security measures such as encryption, access control, and intrusion detection systems are essential to safeguard sensitive data and ensure the reliability of communications within this intricate ecosystem. Addressing these challenges requires innovative approaches that can dynamically adjust to the evolving landscape of IoT-Fog environments [8], ensuring both operational efficiency and robust protection against emerging threats.

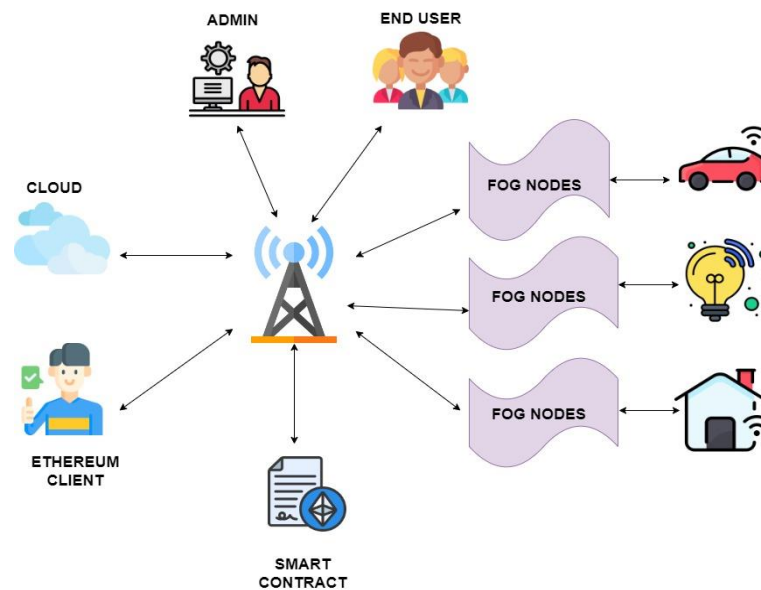


Figure 1. Network model of IoT-Fog computing

The focus of optimization algorithms for attack detection strategy in authenticated networks of IoT-Fog [9]–[11] environments is to improve the efficiency and accuracy of detecting security threats targeting the interconnected devices and infrastructure [2]. This involves leveraging optimization techniques such as genetic algorithms, swarm intelligence, and evolutionary algorithms to optimize parameters, feature selection, and decision-making processes within attack detection systems. The goal is to enhance the detection capabilities [12], reduce false positives, and minimize response times to security incidents.

Furthermore, resource limitations within IoT devices and Fog nodes pose significant obstacles to implementing sophisticated security measures. These devices often operate with constrained computational power and memory, which can limit the feasibility of deploying computationally intensive optimization algorithms for real-time attack detection. Security and privacy concerns are paramount in IoT-Fog environments, where sensitive data is transmitted and processed across multiple nodes and layers, increasing vulnerability to malicious attacks.

Existing solutions for utilizing optimization algorithms in attack detection strategy within authenticated networks of IoT-Fog environments include Swarm Intelligence algorithm. Swarm intelligence algorithms such as ant colony optimization (ACO) and particle swarm optimization (PSO) [13] optimize feature selection [14] and model parameters in attack detection [15], [16] systems, enhancing their performance [17] in identifying security threats. Despite the potential benefits of optimization algorithms in attack detection strategies for authenticated networks of IoT-Fog environments, several challenges and drawbacks exist. They are:

- a. Computational complexity: Optimization algorithms may require significant computational resources, making them unsuitable for resource-constrained IoT devices and Fog nodes.
- b. Lack of scalability: Some optimization algorithms may struggle to scale effectively to accommodate the increasing number of devices and data volumes within the IoT-Fog [12], [18] environment.
- c. Sensitivity to parameters: The performance of optimization algorithms can be sensitive to parameter settings and initialization, requiring careful tuning and optimization.

d. Adaptability to dynamic environments: Optimization algorithms may face challenges in adapting to the dynamic nature of IoT-Fog environments, where network conditions and attack patterns can change rapidly.

In summary, scalability, the heterogeneity of IoT devices, data distribution challenges, resource limitations, and the evolving landscape of security threats all contribute to the complexities of IoT-Fog networks. These issues underscore the critical role of optimization algorithms. They are essential for enhancing the security posture of authenticated IoT-Fog network. These algorithms aim to mitigate vulnerabilities and improve the resilience of IoT-Fog environment against emerging security challenges, thereby safeguarding sensitive data and maintaining operational integrity.

Effective attack detection swiftly identifies and mitigates potential threats to ensure system security and integrity. In the context of attack detection Diro and Chilamkurti [19] recommends leveraging long short-term memory (LSTM) networks for attack detection in Fog-to-things communications. They highlight LSTMs' capability to capture long-term dependencies in sequential data, which improves the accuracy of attack detection by analyzing historical network traffic data. Additionally, they address potential challenges such as computational complexity and dataset requirements. Ethala and Kumarappan [13] introduces a hybrid intrusion detection method for IoT environment, combining spider monkey optimization (SMO) and hierarchical particle swarm optimization (HPSO). These algorithms, inspired by natural behaviors of spider monkeys and hierarchical structures. They address the rising cyber threats in IoT networks by improving the detection accuracy and efficiency through the integrated strengths of SMO and HPSO. The approach includes preprocessing IoT data, optimizing intrusion detection parameters, and implementing the model for real-time network protection. Saeed and Jameel [14] propose a method combining particle swarm optimization (PSO) and a decision tree (DT) classifier for intelligent feature selection in distributed denial of service (DDoS) attack detection. This integration optimizes feature selection from large datasets, enhancing detection accuracy and reducing computational overhead. Daoud and Mahfoudhi [20] introduces secure intelligent method for attack detection (SIMAD), a detection method combining machine learning and anomaly detection to secure IoT-Fog environments by monitoring network traffic and device behavior to identify threats like DDoS and malware with high accuracy. It also highlights challenges such as the need for continuous model updates and managing encrypted traffic in resource-limited settings. Gouda *et al.* [21] proposes an attack detection scheme for mobile ad hoc networks in internet of things (MANET-IoT) environment. This scheme uses the adaptive tunicate swarm algorithm (ATSA) to identify and mitigate threats from blackmailing nodes. This scheme dynamically adapts to network conditions and threat severity. Adrian *et al.* [22] explores using PSO to identify attack points on IoT devices to enhance the efficacy of intrusion prevention systems. PSO inspired by collective animal behaviors, targets the optimization of attack point selected on IoT devices, which are often vulnerable due to limited resources and insufficient security. The study evaluates the effectiveness of this method and discusses potential drawbacks, such as reliability of specific assumptions and performance variability under different conditions.

The motivation for exploring optimization algorithms for attack detection strategy in authenticated networks [23], [24] of IoT-Fog environments stems from the need to address the identified gaps and drawbacks in existing solutions. By investigating the efficacy of optimization techniques in improving the efficiency, scalability, and adaptability of attack detection systems, our study aims to provide valuable insights into mitigating security threats within interconnected ecosystems. Furthermore, by identifying optimal configurations and strategies for leveraging optimization algorithms, we seek to empower stakeholders to develop proactive and effective security measures [25], [26] that uphold the integrity and confidentiality of data in IoT-Fog environment. The contribution of research in attack detection strategy within authenticated networks [27] of IoT-Fog environment lies in its endeavor to bridge the be identified gaps and challenges: i) implementation of authenticated network of IoT-Fog environment; ii) implementation of Sailfish cat optimization algorithm for attack detection strategy in authenticated network of IoT-Fog environment; and iii) comparison of proposed approaches with the existing approaches

Based on the above discussions, the problem is defined as the development of a robust authentication and attack detection system tailored for IoT-Fog environment, which presents significant challenges. The integration of Ethereum smart contracts for device authentication and access control, along with the detection of various types of attacks using a sailfish-cat optimization (SaCO) based generative adversarial network (GAN) approach, aims to enhance security within this complex ecosystem. However, ensuring seamless authentication across decentralized networks and effectively detecting sophisticated attacks remain critical issues. This study addresses these challenges by proposing a hybrid approach that optimizes authentication processes and enhances attack detection mechanisms specifically designed for IoT-Fog environment.

The structure of this paper is outlined as follows: section 2 provides an in-depth exploration of the IoT-Fog network model. SaCO-based GAN framework is described in detail to identify threats in the context of IoT-Fog computing environment. Section 3 contains the findings and conversation about the suggested paradigm. The conclusion of this research work is provided in section 4.

2. METHOD

In this section, proposed attack detection strategy in authenticated network of IoT-Fog environment is presented. The proposed model leverages a hybrid approach combining sailfish-cat optimization with a GAN to detect attacks within an IoT-Fog environment. This approach is preceded by a robust authentication mechanism using Ethereum smart contracts. The process encompasses two primary phases: Authentication and attack detection. During the authentication phase [24], IoT device access is authenticated using access tokens, verified through processes like hashing and encryption. Post-authentication, permission to access data is granted to the user. For attack detection, the process begins with feature selection using Minkowski distance on network data, followed by attack detection using a SaCO-based GAN, which optimally tunes GAN's internal model parameters via integration of sail fish optimizer (SFO) and cat swarm optimizer (CSO) techniques. The authentication and attack detection processes are detailed in Figure 2.

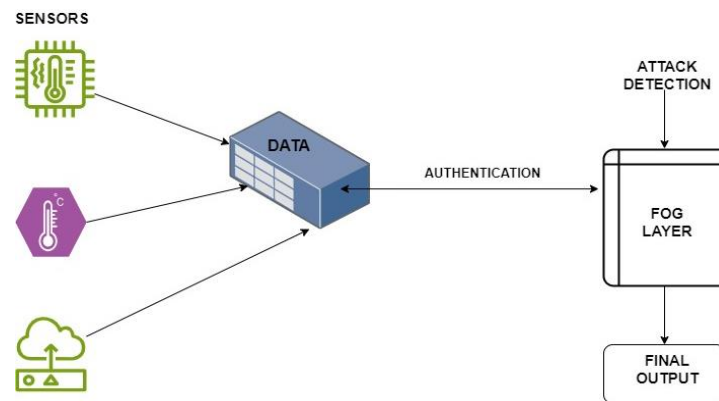


Figure 2. Authentication and attack detection process

2.1. Authentication process

This section describes the authentication process for a decentralized cloud storage system, involving device registration, mapping, authentication, token generation, and data exchange, as depicted in Figure 3. Five entities smart contract, admin, Fog nodes, end user, and IoT devices are involved, with the admin deploying the smart contract to register IoT devices and map them to manage fog nodes. Authentication, access control, and registration are decentralized through the smart contract, which also records approved device users and their encrypted keyword files. The authentication process employed in this study are attributed to the fundamental work proposed in our previous research article [24] where the relevant formulation are thoroughly elucidated.

- a. Device registration phase: The administrator forms a smart contract using the IoT device's ID, Ethereum address, and attributes. They then register it with a user ID and an encrypted password derived from the concatenated device ID and Ethereum address. Secure access is ensured by verifying that the user-entered password matches the stored encrypted one.
- b. Mapping phase: The administrator facilitates IoT device-to-fog node mapping by generating messages from concatenated public keys and Ethereum addresses. Further, concatenate these messages with a random number for the mapping function. After meeting certain conditions, these messages are sent to the respective fog nodes, recorded on IoT devices, and used to secure device-to-user associations. To enhance security, user passwords, IDs, and public keys are hashed and XORed with a security parameter to create a final message stored on IoT devices for user verification.
- c. Authentication phase: When an end user attempts to log into an IoT device, they submit an authentication request to the smart contract, which uses the device's set authentication function to verify their credentials. If the user lacks the required permissions, the smart contract denies access, records the denial, and notifies the user about the rejection.
- d. Token generation phase: The authentication process employs token-based access control, generating tokens from the Ethereum address, device ID, and cryptographic elements. Fog nodes verify these tokens to authenticate user access to IoT devices, with enhanced security provided by digitally signed messages and session keys to secure and verify communications.
- e. Data exchange phase: During the data exchange phase, IoT devices and users securely transmit encrypted data over SSL connections to maintain confidentiality and integrity. Additionally, fog nodes and end users

use private and session keys for encryption. The success of data exchange relies on the consistency of signed messages between fog nodes and end users, ensuring the authenticity and accuracy of the data transferred.

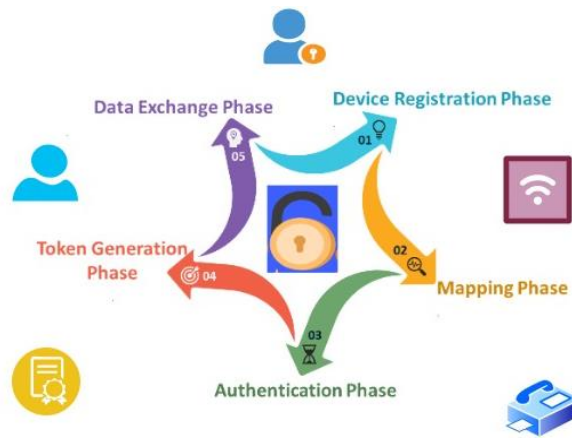


Figure 3. Authentication process

2.2. Attack detection process

Following successful user authentication and data exchange, the attack detection process begins using the SaCO-based GAN approach as illustrated in Figure 4. Initially, input data undergoes feature selection via the Minkowski distance metric to select optimal features. Subsequently, attack detection utilizes the tuned GAN model, optimized by the SaCO, a hybrid of the SFO and CSO. The initial steps of the SaCO-based GAN approach is explained below.

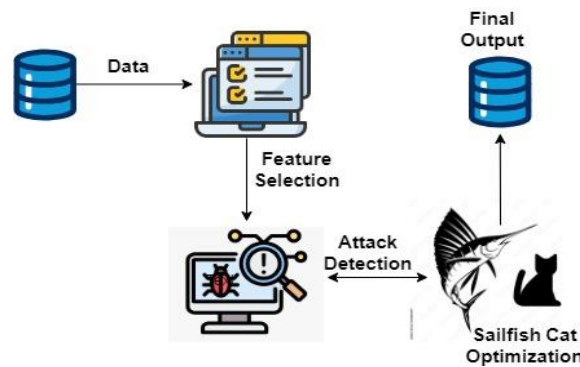


Figure 4. Illustrative representation of the SaCO-based GAN proposed for the detection of attacks in the IoT-Fog environment

2.2.1. Get the input signal

The input data is identified through a systematic analysis of the information extracted from a dataset. This dataset, denoted as Y , consists of n input data points, each providing distinct insights. Analyzing these data points facilitates the discovery of patterns and relationships that are critical for deriving informed conclusions from the dataset.

$$Y = \{C_i; i \in \{1, 2, \dots, n\}\} \quad (1)$$

2.2.2. Minkowski distance for selecting the features

The feature selection module processes the input data C_i by employing the Minkowski distance in its selection algorithm. Specifically, the Minkowski distance of order g between two neighboring data points, denoted as C_j and C_k , quantifies their similarity and is mathematically expressed as (2). This distance metric is essential for determining the relevance of features and enhancing the overall performance of the selection process.

$$B(C_j, C_k) = (\sum_{d=1}^p |C_j^d - C_k^d|^g)^{\frac{1}{g}} \quad (2)$$

2.2.3. Algorithmic procedure of the proposed sailfish-cat optimization for attack detection

The algorithmic procedure of the proposed SaCO for attack detection is meticulously outlined. This approach includes a series of systematic steps that illustrate its functionality and effectiveness. Each step provides a comprehensive understanding of the SaCO operation, supporting efficient attack detection. Refer to Table 1 for a complete list of notations and their descriptions used in the algorithmic procedure of the proposed sailfish-cat optimization for attack detection.

Table 1. Notations and its descriptions

Notations	Description
B_{pos}	The positions of all sailfish
T_{pos}	The positions of all sardine
B_{fit}	Fitness value for each sailfish
T_{fit}	Fitness value for each sardine
Z_{elite}^j	Elite sailfish's
$Z_{injured}^j$	Injured Sardine
$Z_{\tau+1}^j$	New location of sardine j
Z_{τ}^j	Current location of sardine j
Q_s	Prey density
BA	Number of sailfish's attack power at every iteration
M^B	The number of the sailfish in every cycle
M^T	The number of the sardines in every cycle

Step 1. Initialization of the population

In the SFO scheme, sailfish are identified as potential candidates, where each sailfish's position in a search space represents a problem variable. Initially, a population of sailfish is randomly generated in the solution space. The position of the a^{th} member of sailfish during the s^{th} search in b dimensional space is defined as (3):

$$B_{a,s} \in X(a = 1, 2, \dots, d) \quad (3)$$

where b denotes the number of variables and d represents the number of sailfish members. Matrix B_{pos} is utilized to store the positions of all sailfish, representing the variables of all solutions throughout the optimization process.

$$B_{pos} = \begin{bmatrix} B_{1,1} & B_{1,2} & \dots & B_{1,b} \\ B_{2,1} & B_{2,2} & \dots & B_{2,b} \\ \vdots & \vdots & \vdots & \vdots \\ B_{d,1} & B_{d,2} & \dots & B_{d,b} \end{bmatrix} \quad (4)$$

where d indicates the number of sailfish, b denotes the number of variables, and $B_{i,j}$ represents the value of the j^{th} dimension of the i^{th} sailfish. The school of sardines is another important component in the SFO algorithm. It is assumed that the group of sardines also moves within the search space. Therefore, the positions of sardines and their corresponding fitness values are utilized as (5):

$$T_{pos} = \begin{bmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,b} \\ T_{2,1} & T_{2,2} & \dots & T_{2,b} \\ \vdots & \vdots & \vdots & \vdots \\ T_{e,1} & T_{e,2} & \dots & T_{e,b} \end{bmatrix} \quad (5)$$

where e is the number of sardines and $T_{i,j}$ denotes the value of the j^{th} dimension of i^{th} sardine. The T_{pos} matrix signifies the position of all sardines.

Step 2. Determination of fitness

The fitness of each sailfish is determined by evaluating the fitness function as (6):

$$\text{fitness value} = f(B_1, B_2, \dots, B_d) \quad (6)$$

To assess each sailfish, the following matrix displays the fitness values for all solutions:

$$B_{fit} = \begin{bmatrix} f(B_{1,1}) & B_{1,2} & \dots & B_{1,b} \\ f(B_{2,1}) & B_{2,2} & \dots & B_{2,b} \\ \vdots & \vdots & \vdots & \vdots \\ f(B_{d,1}) & B_{d,2} & \dots & B_{d,b} \end{bmatrix} = \begin{bmatrix} F_{B_1} \\ F_{B_2} \\ \vdots \\ F_{B_d} \end{bmatrix} \quad (7)$$

In this scenario, d represents the count of sailfish, B_i , denotes the j^{th} dimension value of the i^{th} sailfish, f computes the fitness function, and B_{fit} records the fitness values representing the fitness or objective function outcome for each sailfish. The initial row of the B_{pos} matrix undergoes evaluation within the fitness function, and its result signifies the fitness value for the respective sailfish within the B_{fit} matrix.

$$T_{fit} = \begin{bmatrix} f(T_{1,1}) & T_{1,2} & \dots & T_{1,b} \\ f(T_{2,1}) & T_{2,2} & \dots & T_{2,b} \\ \vdots & \vdots & \vdots & \vdots \\ f(T_{e,1}) & T_{e,2} & \dots & T_{e,b} \end{bmatrix} = \begin{bmatrix} F_{T_1} \\ F_{T_2} \\ \vdots \\ F_{T_e} \end{bmatrix} \quad (8)$$

where e denotes the number of sardines, $T_{i,j}$ represents the value of the j^{th} dimension of the i^{th} sardine, f signifies the objective function, and T_{fit} stores the fitness value for each sardine. It is important to note that sailfish and sardines are complementary elements in discovering solutions. In this algorithm, sailfish are the primary entities dispersed across the search space, while sardines collaborate to locate optimal positions within this domain. Specifically, sardines may be preyed upon by sailfish during exploration of the search space, prompting sailfish to update their positions upon discovering superior solutions.

Step 3. Elitism

The concept of elitism in the sail fish optimizer (SFO) involves preserving the optimal solution across generations without modification. In the context of group attacks where sailfish target sardines, the elite sailfish's position influences the acceleration and maneuvering of the group. Additionally, wounded sardines, marked in each cycle, become prime candidates for collaborative sailfish hunting. The injured sardine with the highest fitness at a particular iteration and the elite sailfish's coordinates are indicated as Z_{eliteB}^j and $Z_{injuredT}^j$.

Step 4. Attack alternation strategy

Sailfish predominantly engage in solo attacks on prey but can enhance their hunting success by coordinating attacks over time, adjusting their positions relative to other hunters despite no direct coordination. This behavior inspires the SFO, which mimics the sailfish's alternating attack strategy when hunting in groups. In SFO, each sardine adjusts its position based on the optimal location of sailfish and attack intensity to simulate this process effectively.

$$Z_{\tau+1}^j = Z_{eliteB}^j - \beta_j(rand(0,1)) \left(\frac{Z_{eliteB}^j + Z_{injuredT}^j}{2} \right) - Z_{\tau}^j \quad (9)$$

Hence, where $Z_{injuredT}^j$ denotes the position of harmed sardines, where Z_{eliteB}^j denotes the location of elite sailfish, where $rand(0,1)$ denotes the random number between 0 and 1, and where β_j denotes the coefficient at j^{th} iteration. To achieve globally optimal solutions in attack detection, the CSO is incorporated into the algorithm. As a result, the CSO update equation is expressed as (10):

$$Z_{best} = \frac{Z_{\tau}^j(1+a_1l_1)+V_{\tau+1}^j - Z_{\tau+1}^j}{a_1 l_1} \quad (10)$$

The proposed SaCO's equation can be written as (11),

$$Z_{\tau+1}^j = \frac{2a_1l_1}{2a_1l_1+2-\beta_j(rand(0,1))} \left[\frac{Z_{\tau}^j(1+a_1l_1)+V_{\tau+1}^j}{a_1l_1} \left(1 - \frac{\beta_j(rand(0,1))}{2} \right) - \frac{\beta_j}{2} rand(0,1)Z_{injuredT}^j + \beta_j Z_{\tau}^j \right] \quad (11)$$

where, $\beta_j = 2 * rand(0,1) * Q_s - Q_s$. Here, "prey density" Q_s refers to the quantity of prey at each iteration. When the sailfish reduce the amount of prey during group hunting, the Q_s is used to update the sailfish's location around the prey school. The formula for prey density is as (12):

$$Q_s = 1 - \left[\frac{M^B}{M^B + M^T} \right] \quad (12)$$

where, the terms M^B and M^T signifies the number of the sailfish and sardines in every cycle.

Step 5. Pursuit and predation

During the commencement of a hunt, both the sardines' adeptness at evasion and the sailfish's capacity for attack are typically at their zenith. Consequently, in the initial phase of the hunt, sailfish inflict wounds upon the sardines within the prey school without necessarily ensnaring them entirely. The impressive offensive capabilities of sailfish diminish the sardines' efficacy in escaping. These sailfish dynamically adapt their tactics as the sardines accrue injuries, ultimately resulting in a decrease in the prey's ability to evade attacks as the hunt progresses. Subsequently, the success rate of sailfish peaks. Consequently, each sardine adjusts its position utilizing a specific equation tailored to the evolving dynamics of the hunt.

$$Z_{\tau+1}^j = S * (Z_{eliteB}^j - Z_{\tau}^j + BA) \quad (13)$$

The symbols $Z_{\tau+1}^j$ and Z_{τ}^j denote the new and current location of sardine j respectively. The symbol S refers to a random number ranging from 0 to 1. The symbol Z_{eliteB}^j represents the optimal location of elite sailfish. BA denotes the number of sailfish's attack power at every iteration, represented as (14),

$$BA = B * (1 - (2 * itn * \eta)) \quad (14)$$

Here, the coefficients B and η represent factors that decrease the power of the attack value, and itn denotes the current iteration. The number of sardines that update their location during the last phase of the hunt is calculated using BA as (15), (16):

$$\mu = M^T * BA \quad (15)$$

$$\lambda = b_j * BA \quad (16)$$

Here, μ denotes the location of the sardine, while λ represents the number of variables associated with the sardines. The number of variables at iteration j is represented as b_j , and M^T denotes the number of sardines in each cycle. When the BA is less than 0.5, the sardines with μ variables are updated; however, if the BA is greater than or equal to 0.5, then the location of all sardines is updated.

Step 6. Evaluate feasibility

Equation (6) is used to calculate each search agent's fitness. The function that yields the best result is known as the optimum solution. While the value that is highest reflects an optimal fitness measure.

Step 7. Termination

Until the designated iteration is completed or an ideal solution is obtained, the previously indicated procedures continue. The implemented SaCO's pseudo code is provided by Algorithm 1 and flowchart of the proposed model is shown in Figure 5.

Algorithm 1. Pseudo code of the developed model

```

Input: Initial population of sailfish and sardine
Output: Optimal solution (sailfish)
Initialize the populations of sardine and sailfish, along with relevant parameters
Compute the fitness of each individual in the population
Determine the optimal sailfish and sardine based on their fitness
while the stopping criteria are not met:
  for each sailfish:
    Compute the coefficient  $\beta_j$  at the current iteration
    Update the Sailfish Cat Optimization (SaCO) using equation (11)
  end for
  Compute the attack power  $BA$  based on equation (14)
  if  $BA < 0.5$ :
    Compute  $\mu$  based on equation (15)
    Estimate  $\lambda$  based on equation (16)
    Choose the sardine using the value of  $\mu$  and  $\lambda$ 
    Update the location of the selected sardine using equation (13)
  else
    Update the location of all sardines using equation (13)
  end if
  Compute the fitness of all sardine

```



```

    if a better solution is found in the sardine population:
        Update the optimal sailfish with the improved sardine
        Eliminate the hunted sardine
        Update best sailfish and sardine
    end if
end while
Return the best sailfish as the optimal solution.
End

```

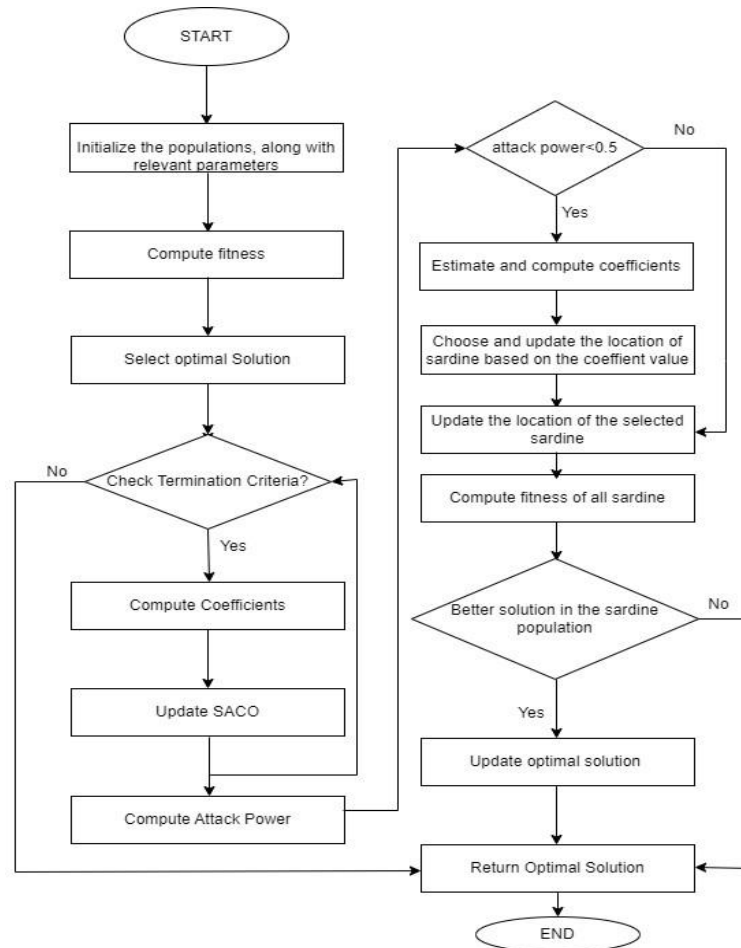


Figure 5. Flowchart of the developed model

3. RESULTS AND DISCUSSION

This section explores the results of the proposed attack detection strategy in the IoT-Fog context, providing details on the experimental setup and dataset description. The SaCO-based generative adversarial network is implemented using Python on a Windows 10 system with an Intel processor and 4 GB of RAM. The strategy utilizes the BoT-IoT database [28], encompassing a diverse array of normal and botnet traffic categorized in CSV, Argus, and PCAP formats, totaling over 72 million records. The dataset consists of PCAP files totaling up to 69.3 GB and CSV flow traffic data of 16.7 GB. It includes various types of attacks such as data exfiltration, denial of service (DoS), DDoS, service scan, keylogging, and OS attacks.

3.1. Comparative techniques

Our proposed technique SaCO based GAN compared with k-neural network (KNN), support vector machine (SVM) [25], neural network (NN) [15], linear regression, deep convolutional neural network (DCNN) [16], and GAN [29], based on recall, precision and F-Measure evaluation metrics. Precision, recall, and F-measure are important metrics to determine algorithm performance because they collectively assess the accuracy in identifying attacks (precision), the completeness of attack detection (recall), and provide a balanced measure that considers both accuracy and comprehensiveness (F-measure), crucial for evaluating the effectiveness of security measures in IoT-Fog environments.

3.2. Comparative analysis

This section provides a detailed comparative analysis of the attack detection model in the IoT-Fog environment, focusing on three distinct types of attacks: backdoor attacks, fuzzers attacks, and DoS attacks. The performance evaluation of the proposed SaCO-based GAN method is assessed using metrics such as precision, recall, and F-measure.

3.2.1. Analysis based on backdoor attack

Figure 6 provides a comparative evaluation of the SaCO-based GAN method specifically developed for detecting backdoor attacks. In Figure 6(a), a comparison of recall is presented, revealing variations with changes in the percentage of training data. Specifically, with 70% training data, existing methods such as KNN, SVM, NN, linear regression, DCNN, GAN, and the developed SaCO-based GAN exhibit recall values of 83.41%, 83.73%, 83.79%, 85.23%, 85.67%, 86.73%, and 87.23%, respectively. Additionally, Figure 6(b) displays a comparative analysis of precision values across different percentages of training data. At a training data percentage of 70%, KNN, SVM, NN, linear regression, DCNN, GAN, and the developed SaCO-based GAN demonstrate precision values of 81.59%, 82.71%, 82.77%, 83.09%, 83.21%, 85.99%, and 87.85%, respectively. Furthermore, Figure 6(c) represents a comparative analysis of F-measure values with varying percentages of training data. The existing attack detection methods (KNN, SVM, NN, linear regression, DCNN, GAN) and the proposed SaCO-based GAN method achieve F-measure values of 82.08%, 82.27%, 82.72%, 84.07%, 84.36%, 85.44%, and 87.07% when trained with 70% of the data. This emphasizes the balanced performance of the SaCO-based GAN in both recall and precision, highlighting its capability to effectively detect backdoor attacks across varying training data scenarios. In summary, Figure 6 demonstrates that the SaCO-based GAN method outperforms traditional methods in recall, precision, and F-measure when detecting backdoor attacks in IoT-Fog environment. The results underscore its potential as an advanced security solution for mitigating backdoor threats effectively.

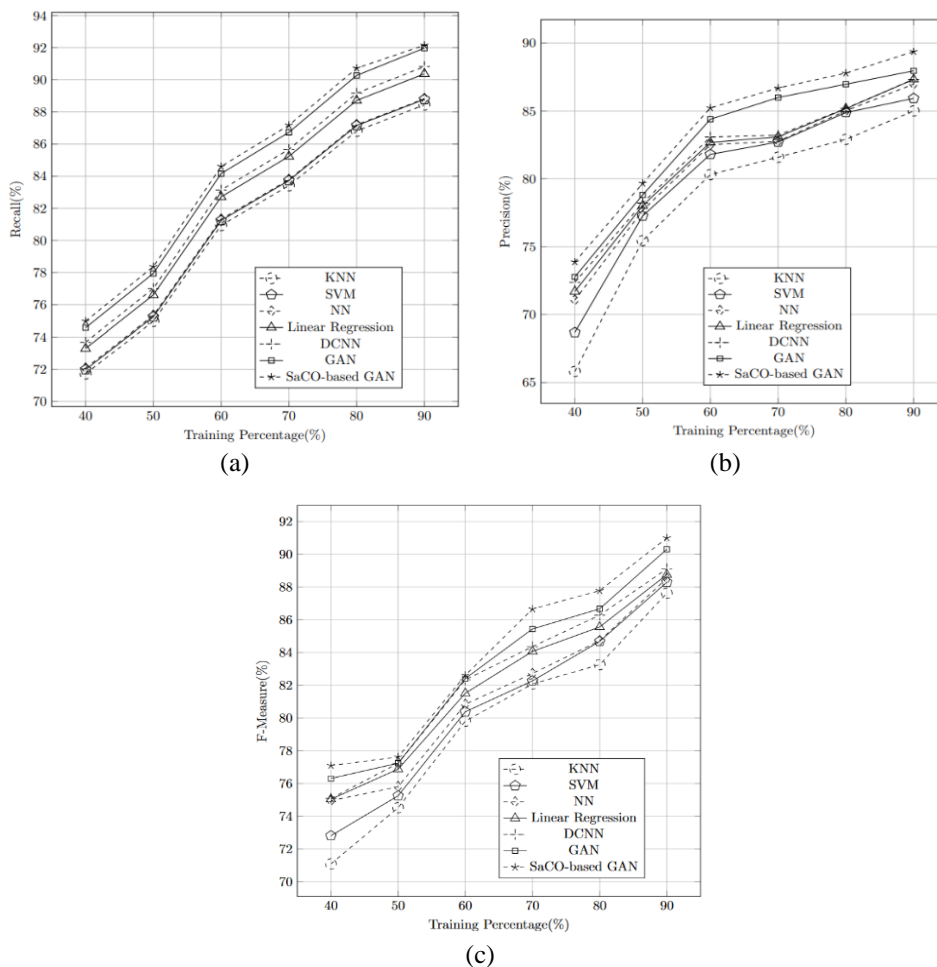


Figure 6. Comparative analysis using backdoor attack (a) recall, (b) precision, and (c) F-measure

3.2.2. Analysis based on DoS attack

Figure 7 provides a comparative analysis of the developed SaCO-based GAN method specifically for detecting DoS attacks. Figure 7(a), displays comparative analysis of recall by altering training data percentage. In 60% of training data, recall attained by existing methods, like KNN, SVM, NN, linear regression, DCNN and GAN and developed SaCO-based GAN are 82.68%, 82.98%, 83.09%, 84.46%, 84.90%, 85.90%, and 86.40%. Figure 7(b) displays comparative analysis of precision value by varying amount of training data percentage. When, training data=60%, precision value obtained by present KNN, SVM, NN, linear regression, DCNN and GAN methods and developed SaCO-based GAN method are 81.72%, 82.46%, 84.16%, 84.18%, 84.46%, 84.75%, and 86.70%. Figure 7(c) represents comparative analysis of F-measure value by changing training data percentage. The existing attack detection methods, such as KNN, SVM, NN, linear regression, DCNN, and GAN and proposed SaCO-based GAN method obtained F-measure of 80.66%, 81.80%, 81.92%, 82.33%, 83.53%, 83.93%, and 84.64% in 60% of training data. This underscores the SaCO-based GAN's balanced performance in recall and precision, further confirming its efficacy in detecting DoS attacks across various training data scenarios. In summary, the SaCO-based GAN method demonstrates superior performance in recall, precision, and F-measure compared to traditional methods, making it a robust solution for enhancing security against DoS attacks in IoT-Fog environments.

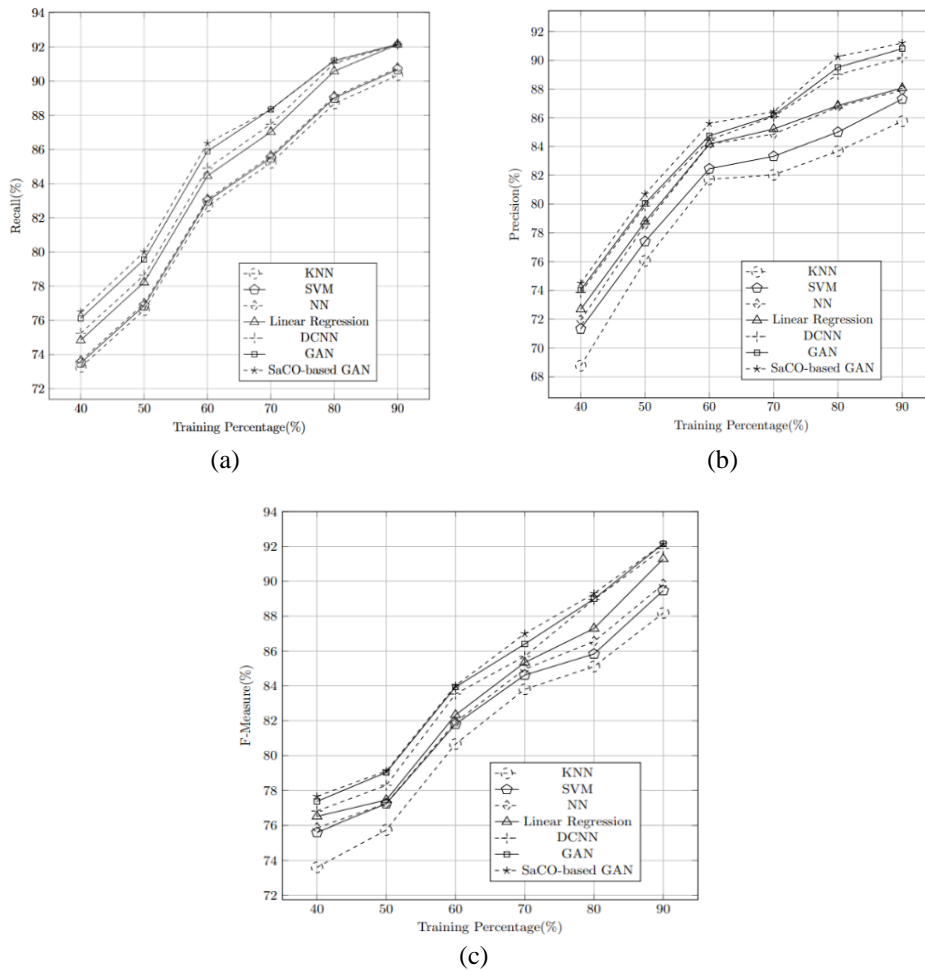


Figure 7. Comparative analysis using DoS attack (a) recall, (b) precision, and (c) F-measure

3.2.2. Analysis based on fuzzers attack

Figure 8 provides a detailed comparative evaluation of the SaCO-based GAN method specifically designed for detecting fuzzers attacks. In Figure 8(a), the analysis of recall is depicted, illustrating variations in training data percentage. Specifically, at a training data percentage of 80%, recall values obtained by existing methods, including KNN, SVM, NN, linear regression, DCNN, GAN, and the developed SaCO-based GAN, are 87.75%, 88.08%, 88.15%, 89.65%, 90.10%, 91.2%, and 91.2%, respectively. Furthermore, Figure 8(b) provides a comparative analysis of precision values across different percentages of training data.

With a training data percentage of 80%, KNN, SVM, NN, linear regression, DCNN, GAN, and the developed SaCO-based GAN exhibit precision values of 83.66%, 85.04%, 86.14%, 86.59%, 88.92%, 89.13%, and 80.25%, respectively. Additionally, Figure 8(c) represents a comparative analysis of F-measure values with varying percentages of training data. The existing attack detection methods (KNN, SVM, NN, linear regression, DCNN, GAN) and the proposed SaCO-based GAN method achieve F-measure values of 84.32%, 84.87%, 85.80%, 86.95%, 88.21%, 88.64%, and 89.3% when trained with 80% of the data. This indicates that the SaCO-based GAN method not only excels in recall and precision but also provides a balanced performance in terms of F-measure, underscoring its effectiveness in detecting fuzzers attacks across varying training data scenarios. These results collectively emphasize the robustness and competitive edge of the SaCO-based GAN method in enhancing security measures against fuzzers attacks in IoT-Fog environment.

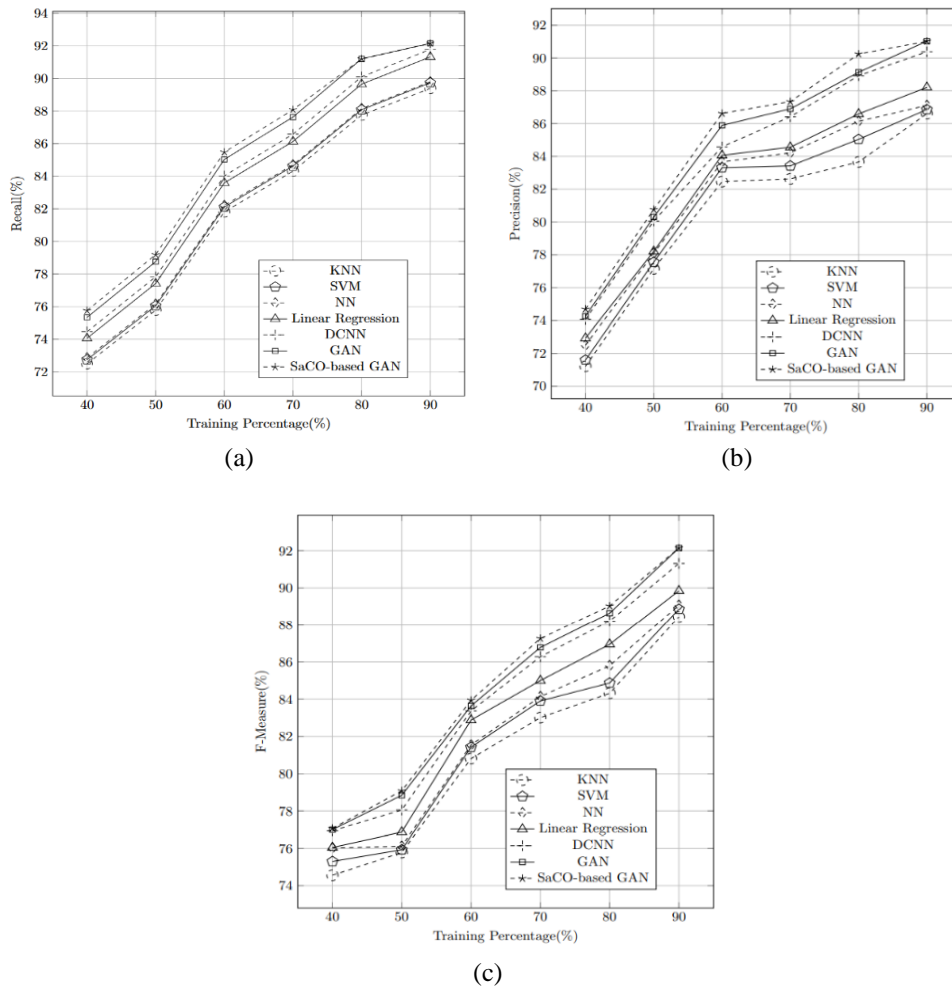


Figure 8. Comparative analysis using fuzzers attack (a) recall, (b) precision, and (c) F-measure

3.2. Comparative discussion

Detailed comparison of the proposed SaCO-based GAN method with existing attack detection schemes, emphasizing recall, precision, and F-measure across different percentages of training data is presented in Table 2. Notably, the SaCO-based GAN achieves an outstanding recall of 92.15%, surpassing established methods such as KNN, SVM, NN, linear regression, DCNN, and GAN, which range from 88.44% to 91.97%. Similarly, in terms of precision, where KNN, SVM, NN, linear regression, DCNN, and GAN range from 85% to 87.96%, the SaCO-based GAN excels with a precision value of 90.64%. Moreover, the F-measure results further underscore its effectiveness, with values ranging from 87.62% to 90.31% for existing methods compared to 91.78% achieved by the SaCO-based GAN. These results highlight the superior performance of the SaCO-based GAN method in accurately detecting attacks, demonstrating its potential for enhancing security in IoT-Fog environments.

Table 2. Comparative analysis

Attack	Metrics (%)	KNN	SVM	NN	Linear regression	DCNN	GAN	SaCO-based GAN
Backdoor	Recall	88.44	88.78	88.82	90.37	90.82	91.97	92.15
	Precision	85	85.93	86.98	87.33	87.33	87.96	90.64
	F-Measure	87.62	88.29	88.53	88.76	89.11	90.31	91.78
DoS	Recall	90.34	90.68	90.76	92.15	92.15	92.15	92.17
	Precision	85.76	87.30	87.92	88.07	90.17	90.81	91.20
	F-Measure	88.18	89.47	89.83	91.28	91.88	92.14	92.15
Fuzzers	Recall	89.39	89.73	89.79	91.32	91.78	92.13	92.15
	Precision	86.6	86.86	87.11	88.21	90.38	91.03	91.20
	F-Measure	88.44	88.86	89.07	89.85	91.38	92.14	92.16

4. CONCLUSION

This research paper introduces a novel approach for detecting attacks in IoT-Fog environment using the SaCO-based GAN method. Authentication involves five entities accessing Ethereum smart contracts, where IoT device access is secured through access tokens, along with subsequent functions like hashing and encryption. Post-authentication, attack detection utilizes Minkowski distance for feature selection and integrates SaCO to optimize GAN parameters. The proposed method achieves impressive performance metrics with high recall, precision, and F-measure values of 92.15%, 91.21%, and 92.16%, respectively, demonstrating its effectiveness in identifying and mitigating attacks in IoT-Fog environment.




REFERENCES

- [1] R. K. Naha, S. Garg, A. Chan, and S. K. Battula, "Deadline-based dynamic resource allocation and provisioning algorithms in Fog-Cloud environment," *Future Generation Computer Systems*, vol. 104, pp. 131–141, Mar. 2020, doi: 10.1016/j.future.2019.10.018.
- [2] M. Kumar, A. Kishor, J. Abawajy, P. Agarwal, A. Singh, and A. Y. Zomaya, "ARPS: an autonomic resource provisioning and scheduling framework for cloud platforms," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, pp. 386–399, Apr. 2022, doi: 10.1109/TSUSC.2021.3110245.
- [3] S. A. Fadhil, "Internet of things security threats and key technologies," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 7, pp. 1951–1957, Oct. 2021, doi: 10.1080/09720529.2021.1957189.
- [4] V. Tomer and S. Sharma, "Detecting IoT attacks using an ensemble machine learning model," *Future Internet*, vol. 14, no. 4, Mar. 2022, doi: 10.3390/fi14040102.
- [5] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres, and I. Kantzavelou, "IoT: communication protocols and security threats," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 1–13, 2023, doi: 10.1016/j.iotcps.2022.12.003.
- [6] S. Bharadwaj, P. Amin, D. J. Ramya, and S. Parikh, "Reliable human authentication using AI-based multibiometric image sensor fusion: assessment of performance in information security," *Measurement: Sensors*, vol. 33, Jun. 2024, doi: 10.1016/j.measen.2024.101140.
- [7] A. A. Abba Ari *et al.*, "Enabling privacy and security in cloud of things: architecture, applications, security and privacy challenges," *Applied Computing and Informatics*, vol. 20, no. 1/2, pp. 119–141, Jan. 2024, doi: 10.1016/j.aci.2019.11.005.
- [8] E. Huaranga-Junco, S. González-Gerpe, M. Castillo-Cara, A. Cimmino, and R. García-Castro, "From cloud and fog computing to federated-fog computing: a comparative analysis of computational resources in real-time IoT applications based on semantic interoperability," *Future Generation Computer Systems*, vol. 159, pp. 134–150, Oct. 2024, doi: 10.1016/j.future.2024.05.001.
- [9] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11041–11052, Nov. 2020, doi: 10.1109/JIOT.2020.2993410.
- [10] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for Fog-driven IoT healthcare system," *Security and Communication Networks*, vol. 2021, pp. 1–16, Jan. 2021, doi: 10.1155/2021/6658041.
- [11] S. Xu, J. Ning, J. Ma, X. Huang, H. H. Pang, and R. H. Deng, "Expressive bilateral access control for internet-of-things in cloud-fog computing," in *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, Jun. 2021, pp. 143–154, doi: 10.1145/3450569.3463561.
- [12] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of things attack detection using hybrid deep learning model," *Computer Communications*, vol. 176, pp. 146–154, Aug. 2021, doi: 10.1016/j.comcom.2021.05.024.
- [13] S. Ethala and A. Kumarappan, "A hybrid spider monkey and hierarchical particle swarm optimization approach for intrusion detection on internet of things," *Sensors*, vol. 22, no. 21, Nov. 2022, doi: 10.3390/s22218566.
- [14] A. A. Saeed and N. G. M. Jameel, "Intelligent feature selection using particle swarm optimization algorithm with a decision tree for DDoS attack detection," *International Journal of Advances in Intelligent Informatics*, vol. 7, no. 1, Mar. 2021, doi: 10.26555/ijain.v7i1.553.
- [15] D. Erhan and E. Anarim, "Hybrid DDoS detection framework using matching pursuit algorithm," *IEEE Access*, vol. 8, pp. 118912–118923, 2020, doi: 10.1109/ACCESS.2020.3005781.
- [16] M. Sahidullah *et al.*, "Introduction to voice presentation attack detection and recent advances," *Handbook of Biometric Anti-Spoofing. Advances in Computer Vision and Pattern Recognition*, 2023, pp. 339–385.
- [17] S. Joshi *et al.*, "Unified authentication and access control for future mobile communication-based lightweight IoT systems using blockchain," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/8621230.
- [18] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things*, vol. 19, Aug. 2022, doi: 10.1016/j.iot.2022.100564.
- [19] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in fog-to-things communications," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124–130, Sep. 2018, doi: 10.1109/MCOM.2018.1701270.
- [20] W. Ben Daoud and S. Mahfoudhi, "SIMAD: Secure intelligent method for IoT-Fog environments attacks detection," *Computers, Materials and Continua*, vol. 70, no. 2, pp. 2727–2742, 2022, doi: 10.32604/cmc.2022.020141.
- [21] H. A. Gouda, M. A. Ahmed, and M. I. Roushdy, "Optimizing anomaly-based attack detection using classification machine learning," *Neural Computing and Applications*, vol. 36, no. 6, pp. 3239–3257, 2024, doi: 10.1007/s00521-023-09309-y.




- [22] R. Adrian, A. J. Okke, M. A. R. Somardani, and T. Widiyarsi, "Determination of attack points on IoT devices based on particle swarm optimization to support intrusion prevention system," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2022, pp. 47–50, doi: 10.1109/ISRITI56927.2022.10053073.
- [23] M. S. Eddine, M. A. Ferrag, O. Friha, and L. Maglaras, "EASBF: an efficient authentication scheme over blockchain for fog computing-enabled internet of vehicles," *Journal of Information Security and Applications*, vol. 59, Jun. 2021, doi: 10.1016/j.jisa.2021.102802.
- [24] K. N. Pallavi and V. Ravi Kumar, "Authentication-based access control and data exchanging mechanism of IoT devices in fog computing environment," *Wireless Personal Communications*, vol. 116, no. 4, pp. 3039–3060, Feb. 2021, doi: 10.1007/s11277-020-07834-w.
- [25] F. Siddiqui, J. Beley, S. Zeadally, and G. Braught, "Secure and lightweight communication in heterogeneous IoT environments," *Internet of Things*, vol. 14, Jun. 2021, doi: 10.1016/j.iot.2019.100093.
- [26] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, vol. 15, no. 9, pp. 1200–1215, Oct. 2021, doi: 10.1080/17517575.2020.1712746.
- [27] F. Zerrouki, S. Ouchani, and H. Bouarfa, "PUF-based mutual authentication and session key establishment protocol for IoT devices," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 12575–12593, 2023, doi: 10.1007/s12652-022-04321-x.
- [28] N. Moustafa, "The Bot-IoT dataset," *IEEE Datapor.* 2019.
- [29] S. Song, T. Mukerji, and J. Hou, "Geological facies modeling based on progressive growing of generative adversarial networks (GANs)," *Computational Geosciences*, vol. 25, no. 3, pp. 1251–1273, Jun. 2021, doi: 10.1007/s10596-021-10059-w.

BIOGRAPHIES OF AUTHORS



Pallavi Kanthamangala Niranjana    earned her bachelor's degree in information science and engineering from KVG College of Engineering, Sullia, in 2005, followed by her master's degree in computer science and engineering from NMAM Institute of Technology, Nitte, in 2008. She completed her Ph.D. in computer science and engineering titled "Fog based data security for communication between cloud and internet of things" from Visvesvaraya Technological University, Karnataka, India in 2022. Currently, she serves as an associate professor in the Department of Computer Science and Engineering at NMAM Institute of Technology, Nitte, Karkala. Her research interests include internet of things, machine learning, deep learning, and network security. She can be contacted at email: pallavi@nitte.edu.in.



Ravikumar Venkatesh    earned his bachelor's degree in electrical and electronics Engineering from Kuvempu University, Karnataka, India, followed by his master's degree in computer science and engineering from Visvesvaraya Technological University, Karnataka, India. He completed his Ph.D. in computer science and engineering, titled "Study of feature selection using statistical and semantic phrase techniques with text document," also from Visvesvaraya Technological University, Karnataka, India. Currently, he holds the position of professor and head in the Department of Information Science and Engineering at Vidyavardhaka College of Engineering, Mysuru. His research interests encompass artificial intelligence, natural language processing (NLP), blockchain technology, cloud computing, big data, data mining, and internet of things. He can be reached at email: ravikumarv@vvce.ac.in.