

# 11 11

# 11

 Quick Submit Quick Submit IBD Institute

---

## Document Details

### Submission ID

trn:oid::1:2970825082

### Submission Date

Jul 23, 2024, 5:14 PM GMT+3

### Download Date

Jul 23, 2024, 5:21 PM GMT+3

### File Name

final\_draft.docx

### File Size

812.1 KB

13 Pages

8,523 Words

49,571 Characters



## 21% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

**Caution: Review required.**

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

### Detection Groups

- 
**1 AI-generated only 15%**  
 Likely AI-generated text from a large-language model.
- 
**2 AI-generated text that was AI-paraphrased 6%**  
 Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

#### Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

### Frequently Asked Questions

#### How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (\*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

#### What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



# Chatterjee Correlation-Driven Feature Selection with Artificial Bee Colony for Residual Neural Network-based Intrusion Detection in IoT Networks

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1<sup>st</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

line 1: 2<sup>nd</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

line 1: 3<sup>rd</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

line 1: 4<sup>th</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

line 1: 5<sup>th</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

line 1: 6<sup>th</sup> Given Name Surname  
line 2: dept. name of organization  
(of Affiliation)  
line 3: name of organization (of  
Affiliation)  
line 4: City, Country  
line 5: email address or ORCID

**Abstract**—IoT network intrusion detection presents a number of difficulties, necessitating the skillful use of network device attributes for precise threat identification. With a multi-phase approach, this research provides a novel intrusion detection method. The proposed method utilizes Chatterjee correlation's unique ability to capture nonlinear interactions. The feature selection process is optimized using the artificial bee colony (ABC) algorithm. A customized residual neural network (ResNet) is then presented, which is designed to extract complex associations from the data while reducing computational burden. The suggested ResNet design consists of three stacks of residual blocks, which are carefully set up to learn both low- and high-level features. Feature extraction is improved by progressively increasing the count of filters inside each layer. The first convolutional layers are critical in shaping the learning process as they extract pertinent characteristics and maintain spatial information. To ensure the effectiveness of the network in collecting complex relations, a method for converting 1D features into 2D images is provided to facilitate ResNet input. Experimental results indicate that the suggested approach is exceptionally adept at identifying intrusions in the Internet of Things (IoT) networks, achieving accuracy of 98.5711% 1The combination of artificial bee colony optimization, ResNet, and Chatterjee correlation leads in a strong and effective intrusion detection system that shows promise for improving the security of Internet of Things environments.

**Keywords**—Intrusion Detection, IoT Networks, Chatterjee Correlation, Artificial Bee Colony, Residual Neural Network, Feature Selection, Network Security

## I. INTRODUCTION

IoT technologies have gained extensive adoption in a variety of fields by enabling autonomous connectivity between devices. By 2030, there is expected to be 50 billion IoT devices, with uses ranging from smart cities to healthcare [1],[2]. Nevertheless, the intricate integration of IoT devices on wireless networks, frequently in unsupervised situations, has created vulnerabilities that can be exploited by malevolent actors, bringing with it hitherto unheard-of security concerns. Breaking into data or systems without authorization can lead to severe outcomes, such as damaging the reliability of information and endangering the well-being of people who depend on essential applications [3].

A number of dynamic dangers can compromise the security and performance of modern networks. Various types of attacks, such as clever phishing schemes, malware infections, and DDoS assaults, pose serious risks to the integrity and operation of systems. The challenge is even greater in the area of the IoT, where devices operate in different settings with distinct connection protocols. These attacks are faster and more complicated than what traditional intrusion detection methods can handle, demanding flexible and context-sensitive systems that can effectively detect and prevent different kinds of intrusions.

In this situation, intrusion detection systems, or IDSs, are promising tools for monitoring Internet of Things environments at the network level. These technologies offer real-time data packet analysis, which can assist in detecting and preventing malicious activity. However, implementing IDSs in the IoT domain also includes several challenges. These include the

necessity to function in demanding environments with low energy, limited processing power, quick reaction times, and the difficult chore of managing enormous amounts of data. A thorough grasp of the inherent security flaws in IoT systems is necessary for the ongoing and vital research effort to improve embedded IDSs for IoT [4]. In this regard, many studies have focused on this area:

Paper [5] investigates adversarial challenges in Network Intrusion Detection Systems (NIDS) by employing Deep learning (generative adversarial networks) with particle swarm optimization (PSO) and genetic algorithms (GA). The study assesses on NSL-KDD and UNSW-NB15 datasets. Research [6] aims to enhance IDS efficiency through the Neural Networks, Random Forest, and SVM, with evaluation conducted on standard datasets such as kddcup 99. A deep blockchain framework is introduced in [7] for secure distributed intrusion detection and privacy in IoT networks, utilizing a bidirectional long short-term memory algorithm for intrusion detection, assessed on UNSW-NB15 and Bot-IoT. A Hybrid IDS (HIDS) for groups is introduced in [8], leveraging a combination of a C5 and SVM classifier for enhanced protection of IoT devices. The model analyzed on the Bot-IoT featuring diverse attacks and legitimate IoT network traffic. A new breach detection system based on misuse integrated classification-based model is introduced in [9], focusing on detecting five groups (Exploit, DOS, Probe, Generic, and Normal), utilizing the UNSW-NB15 dataset for model development. A comprehensive study is conducted in [10] on classifiers to advance anomaly-based IDSs, evaluating their effectiveness with reference to the CIDDs-001, UNSW-NB15, and NSL-KDD. Statistical analyses are utilized to detect significant differences between classifiers, such as the Friedman and Nemenyi tests. while The classifiers' performance on IoT-dedicated hardware is assessed through the utilization of Raspberry Pi, examining the time it takes for the classifiers to respond. An IDS is introduced in [11], employing an advanced DCNN for car CAN bus protection. The DCNN, tailored for CAN bus data traffic, autonomously learns network patterns, achieving high detection performance, and is evaluated on datasets created from real vehicle scenarios. A mechanism for combining and stacking features across multiple dimensions, termed Multi-dimensional Feature Fusion and Stacking Ensemble Mechanism is proposed in [12] for effective detection of abnormal behaviors, utilizing multiple basic feature datasets derived from diverse aspects of traffic information. Experimental results is performed on datasets KDD Cup 99, NSL-KDD, UNSW-NB15, and CIC-IDS2017. A Federated Learning scheme is proposed in [13] utilizing local training and inference to protect data privacy in IoT intrusion detection. Updates from the devices are sent to a distant server, which compiles and disseminates an improved detection model, and evaluated on an NSL-KDD to assess its efficiency. GA-based Feature Selection (GbFS), an improved GA for feature selection technique, is introduced in [14]. The method, incorporating parameter tuning and a novel fitness function, is analyzed on CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT. A CNN intrusion approach is developed in [15]. This DL-based model specifically targeting DoS attacks, utilizing the widely used KDD CUP 1999 and the more advanced CSE-CIC-IDS2018 dataset. In [16], a novel anomaly detection method called MDS\_AD is presented. It combines PCA, isolation forest,

and locality-sensitive hashing (LSH) approaches. to effectively address challenges in anomaly detection, particularly for multiaspect data. The approach is validated via experiments undertaken on the UNSW-NB15 dataset. Research [17] introduces an algorithm that leverages a Genetic Algorithm (GA) in combination with 5-fold cross-validation to identify the bagging classifier and optimize the structure of a CNN model for impactful feature extraction. Article [18], deploys logistic regression (LR), naive Bayes (NB), and decision tree (DT) with a voting classifier, exhibiting greater accuracy in comparison to current cutting-edge methods. The proficiency of the suggested method is examined using the CICIDS2017 dataset. The suggested approach in [19] addresses the need for a well-organized classification methodology by applying ML approaches, specifically SVM and NB, on the NSL-KDD. In [20], a model is created, combining enhanced random forest (IRF) techniques with enhanced genetic algorithm and PSO (EGA-PSO) for feature selection and classification. The model is examined on the NSL-KDD. ID Tree (IntruDTree) A security concept based on machine learning is presented in [21]. The model demonstrates effectiveness in prediction accuracy and computational complexity reduction by minimizing feature dimensions, evaluated on cybersecurity datasets with criteria such as precision, recall, fscore, accuracy, and ROC values. A flexible Intrusion Detection System (IDS) that is efficient developed in [22] using a deep neural network (DNN), exploring various datasets generated through static and dynamic approaches to identify the optimal algorithm for detecting future cyberattacks. Experiments run on KDDCup 99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017. A ML-based IDS is suggested in [23] for detecting IoT attacks, utilizing ML-supervised algorithms on the UNSW-NB15. The paper employs feature scaling, normalization, and Principal PCA dimensionality reduction. In [24], four feature subsets taken from the NSLKDD are used for intrusion detection using SVM, K-nearest neighbor, LR, NB, MLP, RF, Extra-tree classifier (ETC), and DT. Paper [25] evaluates the influence of hyperparameters on the results of ANNs for ID, conducting examinations on the NSL-KDD and CICIDS2017. An intrusion detection model is proposed in [26], utilizing machine learning to identify cyberattacks in Internet of Things networks with limited resources. The model is improved by sampling, eliminating multicollinearity, and testing on the CICIDS2017 and NSL-KDD datasets. A smart intrusion detection system (IDS) named Passban is presented in [27]. as a method that can safeguard IoT devices that are directly connected. This model demonstrates detection of several kinds of malicious traffic, including SYN flood assaults, HTTP and SSH brute force attacks, and port scanning. In [28], deep learning architectures are put out as a means of creating a robust and adaptable network intrusion detection system (IDS), with a focus on their capacity to identify both known and novel network behavioral characteristics. The UNSW-NB15 is used to illustrate the method's efficacy. Article [29] suggests a new way of finding and stopping attacks on a Blockchain system by using data fusion and clustering features. The method uses an AI model to check and group data in Blockchain networks, and a math model to merge data. Article [30] shows how to make a better network attack detector by using a DNN and Q-learning, a kind of reinforcement learning. The detector can learn by itself and

adjust its settings automatically. It can find and stop many kinds of attacks on the NSL-KDD.

The use of ML algorithms and feature selection approaches has been the main focus of current research on the detection of intrusions on Internet of Things networks. Despite significant progress, a crucial gap still exists in addressing complex relationships among features, especially when it comes to the existence of no linear dependencies. Our article tries to fill this gap by introducing the Chatterjee correlation coefficient, a measure that can find nonlinear connections, and using it in an ABC optimization technique for better feature selection. The goal is also to lower the amount of computing needed by other approaches and to find complicated patterns in data from Internet of Things networks more effectively by using a special ResNet.

Four different stages of development will take place in the proposed approach. The Chatterjee correlation coefficient, which indicates relevant characteristics needed to detect intrusions, shall be used in the initial selection of features. In order to ensure a well balanced and usable subset of features, the feature selection process is then optimized by means of an Artificial Bee Colony method. After this, a method to convert selected 1D features into 2D images and optimize their input so that it can be used in the next application of your custom ResNet is implemented. The final phase involves the ResNet-based classification, where the network's architecture is strategically configured to capture intricate relationships within the feature space.

The rest of this essay can be structured in the subsequent manner: In next Section, we explore the dataset and offer an in-depth analysis of the dataset that we used in our research. The technique, serves as the foundation for our article, is presented in Section 3 and includes a breakdown of every stage of our unique intrusion detection system. In Section 4, we go over the metrics and criteria that were utilized to assess how well our suggested intrusion detection system performed. Section 5 is dedicated to the simulation results, where we introduce and delve into the empirical results, showcasing the effectiveness of our method. Section 6 involves the comparison, providing a thorough analysis of our proposed methodology against existing intrusion detection approaches. Lastly, Section 7 wraps up the work by reviewing the ramifications and summarizing the major discoveries, and proposing avenues for future research.

## II. DATASET

### A. NSL-KDD dataset

The NSL-KDD was first introduced and published in [31]. The authors proposed this dataset as an enhanced iteration of the initial KDD Cup 99, which had been widely used but was recognized to have certain limitations. In order to overcome these drawbacks and offer a more demanding and realistic benchmark for assessing IDS, the NSL-KDD was developed. It includes important records from the entire KDD dataset and provides scholars with downloadable files. In order to ensure impartiality, redundant records shall be removed in a systematic manner. In order to carry out thorough tests, this creates a logical distribution of records between train and test data sets.

The difficulty levels of the records are taken into account during the dataset selection process in inverse proportion to the record percentages in the original KDD dataset. Every NSL-KDD record has 41 attributes that describe different flow features and a label that either indicates an offensive mode or typical behavior. One normal class and one attack class, comprising the four attack types DoS, Probe, R2L, and U2R, are captured by the 42nd attribute. This dataset's structure enhances its applicability for realistic network security scenario assessments, fostering the development and evaluation of intrusion detection models [32].

### B. Preprocessing Stages

In the preprocessing phase, we employed three essential techniques to enhance the quality of our dataset.

- Min-Max Normalization: We applied this to standardize the values within a specified scale, making sure that all numerical attributes participate uniformly in the examination. The normalization is achieved using the following Equation 1:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

In this equation  $X_{min}$  and  $X_{max}$  depict the lowest and highest values, respectively, and  $X$  represents the initial feature value. This process scales the values between 0 and 1.

- Data Cleaning with K-Nearest Neighbors (KNN): To handle missing data, we employed the K-Nearest Neighbors (KNN) imputation. KNN imputation predicts absent values through considering the amounts of their nearest neighbors. In the context of our study,  $k = 4$  implies that the algorithm identifies the four nearest data points to the one with missing values and averages or combines their values to impute the missing entry. When determining the closest neighbors, the Euclidean distance serves as a widely employed metric to gauge the separation between two points  $(x, y)$ . Its Formula is shown in Equation 2.

$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (2)$$

- Data Partitioning with the Holdout approach: We used the holdout approach to divide the dataset into two halves: 30% was set aside for testing and the rest for training. The holdout approach divides the data into two distinct groups at random, one for model training and the other for performance evaluation. This division makes our intrusion detection method more reliable by ensuring an objective evaluation of the model's generalization skills on untested data.

## III. METHODOLOGY

This section begins by giving a thorough analysis of the fundamental algorithms and ideas that are necessary to lay the groundwork for the suggested method. Thereafter, a comprehensive and detailed description of the proposed approach is provided in addition to all its minute details.



### A. Chatterjee Correlation Coefficient

This **technique** which was recently introduced in correlation analysis, is useful for the analysis of nonlinear interactions in datasets. Examine a dataset  $(X, Y)$  with  $Y$  being non-constant, represented as  $(x_1, y_1), \dots, (x_n, y_n)$ .

To compute the Chatterjee correlation coefficient, we assume ordered inputs as  $(x_{(1)}, y_{(1)}), \dots, (x_{(n)}, y_{(n)})$ , arranged such that  $x_{(1)} \leq \dots \leq x_{(n)}$ . If set  $X$  has no duplicate data, then a unique state is created.

The parameter  $r_i$ , or rank, is determined as the count of  $j$  for each  $i$  where  $y_{(j)} \leq y_{(i)}$ . Chatterjee correlation coefficient in this scenario is calculated using (3):

$$\xi_n(X, Y) = 1 - \frac{3 \sum_{i=1}^{n-1} |r_{i+1} - r_i|}{n^2 - 1} \quad (3)$$

$n$  is the quantity of **instances** in each set of  $X$  and  $Y$ .

In cases where duplicate data exists in the  $X$ -set, one of the conditions, in which the  $x_i$ s are arranged randomly in increments the same as the previous explanation, is considered. Assuming the earlier definition for  $r_i$ , the parameter  $l_i$  is established as the count of  $j$  where  $y_{(j)} \geq y_{(i)}$ . The Chatterjee correlation coefficient is then defined as (4):

$$\xi_n(X, Y) = 1 - \frac{n \sum_{i=1}^{n-1} |r_{i+1} - r_i|}{2 \sum_{i=1}^n l_i (n - l_i)} \quad (4)$$

where  $l_i$  is the new parameter defined for the presence of duplicate data. These formulations provide a robust means to compute the Chatterjee correlation coefficient, accommodating both scenarios of dataset configurations [33].

### B. Artificial Bee Colony Optimization Algorithm

The behavior of individual insects is governed by a few basic rules that support the self-organization of bees, particularly evident in complex tasks such as the organized collection and processing of nectar. Bees decide to forage based on dances performed by nestmates, creating a structured dance floor area for communication. Upon reaching a nectar source, a bee may either abandon, continue foraging, or recruit nestmates through dancing. The decision is probabilistic [34].

To gain insights into the fundamental behavioral traits of foragers, let's analyze Fig. 1. Imagine **two recognized food supplies have been identified**, A and B. Initially, a prospective forager is an unemployed bee with no knowledge of nearby food sources. This bee faces two options: it can act as a scout, spontaneously exploring the surroundings for food based on internal motivation or external cues (denoted as S in Fig. 1), or it can become a recruit by observing waggle dances and then search for a food source (denoted as R in Fig. 1). Once a food source is found, the bee memorizes its location and transforms into an employed forager.

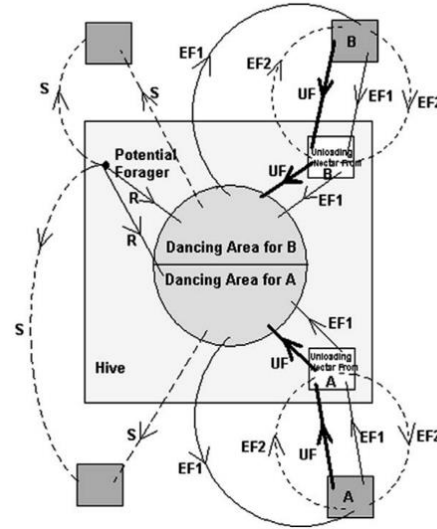


Fig. 1. fundamental behavioral traits of bees searching for nectar [33].

After returning to the hive and unloading the nectar, the forager bee has several choices: it may become an uncommitted follower by abandoning the food source (UF), Before returning to the same food source, dance and recruit nestmates (EF1), or carry-on foraging without recruiting (EF2). Interestingly, not every bee starts foraging at the same time. Research verifies that fresh bees start foraging in proportion to the distinction in the overall quantity of bees that will eventually exist and the population of bees that are currently foraging [35]. The ABC algorithm, developed by Karaboga in 2005, draws inspiration from the foraging **conduct** of honeybees as they seek nourishment (nectar) for their colony. This swarm-based metaheuristic algorithm is designed for optimizing NP-hard problems, relying on the principles of **autonomous coordination and task specialization** to achieve swarm intelligence. The artificial bee colony consists of worker bees attached to particular food sources, observer bees watching dances to identify a food source, and scout bees haphazardly seeking for food. Initially, scouts locate available food sources, and then utilized and onlooker bees exploit the nectar. As bees become exhausted, the employed bees may turn into scouts **to seek additional nourishment** positions. In the **technique**, the place of a **nutritional reservoir shows** a feasible **answer**, and the nectar quantity **matches with** the solution's **grade**. Initialization, employed bee phase, spectator bee phase, and scout bee phase are the four stages of the ABC **technique** [36].

1. Initialization: The ABC begins by **haphazardly** selecting potential solutions (food sources) for employed bees. The solutions are generated using the (5).

$$x_{i,j} = x_j^{min} + \mu(x_j^{max} - x_j^{min}) \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, D \quad (5)$$

In which  $x_{i,j}$  is the  $j^{\text{th}}$  dimension of the  $i^{\text{th}}$  employed bee,  $\mu$  is a **randomized digit** in  $[0, 1]$ ,  $D$  is the **the dimensionality of the issue**, and  $N$  is the size of the

swarm. Every used bee's abandonment counter (AC) gets reset.

2. Employed Bee Phase: Each used bee produces a new candidate solution by updating one parameter using the (6).

$$v_{i,j} = \psi x_{i,j} + \phi(x_{i,j} - x_{r,j}) \quad (6)$$

Here,  $\psi$  is unity,  $x_{r,j}$  is a randomly selected candidate in the neighborhood,  $i$  and  $r$  are distinct members from the same collection.  $\phi$  is a random number in  $[-1, 1]$ . Fitness values are calculated using (7).

$$fit_i = \begin{cases} \frac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{otherwise} \end{cases} \quad (7)$$

in which  $f_i$  is the objective function value. If the fresh answer improves fitness, it substitutes the existing solution, and the AC is reset; otherwise, the AC is improved.

3. Onlooker Bee Phase: Onlooker bees select used bees based on fitness probabilities determined by Roulette wheel selection. Using the same equation as in the employed Bee Phase, the answer provided by the selected employed bee is enhanced. If the new solution's fitness is superior, the onlooker bee replaces the employed bee, resetting the AC; otherwise, the AC is increased. The likelihood of choosing the  $i^{\text{th}}$  employed bee is determined through the process of Roulette wheel selection using the (8).

$$p_i = \frac{fit_i}{\sum_{j=1}^N fit_j} \quad (8)$$

4. Phase of the Scout Bee: After checking the hired bees' abandonment counters, those who surpass a certain threshold are turned into scout bees. Scout bees use the initialization equation to come up with fresh solutions. The scout bee turns into an employed bee once more after the abandonment counter is reset. preventing population stagnation [37].

### C. Residual Neural Network

Over the course of time, significant advancements in image recognition and classification have been achieved through the progression of deep convolutional neural networks [38]. As networks delve into more intricate tasks, challenges such as the vanishing gradient problem arise. In the case of excessively deep networks, the ability to learn even simpler problems may be compromised. The degradation problem is the result of a model's accuracy gradually declining when its layers are continuously increased to a point where it saturates [39]. In response to this problem, He and colleagues were the first to discover and offer a significant remedy that allowed for the training of models with more than 2000 layers and improved accuracy. The groundbreaking creation of a model intended to aid in the training of neural networks with more depth is marked by the introduction of a residual learning framework in [40]. Rather of acquiring unconnected functions, the layers are explicitly redefined to grasp residual functions concerning the

inputs to the layer. The proposed framework addresses the inherent issues linked to training deeper networks. The empirical evidence presented in the paper comprehensively demonstrates that these residual networks are not only easier to optimize but also exhibit enhanced accuracy with substantially increased depth.

The ResNet architecture was devised to address challenges encountered in dl training, which is generally time-consuming and constrained with regard to the quantity of layers. ResNet uses skip connections, or shortcuts, to address the complexities present in deep learning. In contrast to alternative architectural models, ResNets demonstrate a noteworthy advantage: their performance remains consistent while the framework becomes more intricate, computational calculations are also streamlined, leading to improved network training capabilities. The ResNet model achieves this by incorporating skip interconnections across 2 to 3 layers, incorporating ReLU and batch normalization within the structures [41].

A residual network is comprised of residual blocks, as depicted in Fig. 2, illustrating the structure of such a block. Within this figure, certain layers utilize skip connections. Consider a neural network with an input denoted as  $x$ , approximating  $H(x)$ . The differences between them is expressed as  $R(x)$ , specified by the (9) Equation.

$$R(x) = H(x) - x \quad (9)$$

$R(x)$  serves as a residual function, akin to proposing that several nonlinear layers can progressively approximate intricate functions by also suggesting their ability to asymptotically approximate residual functions. The objective of the layers within the residual block is to unveil the residual function  $R(x)$ , leading to the (10).

$$F(x) = H(x) - x \quad (10)$$

Consequently, the primary function is declared as  $F(x) + x$ , as evident in Fig. 2.

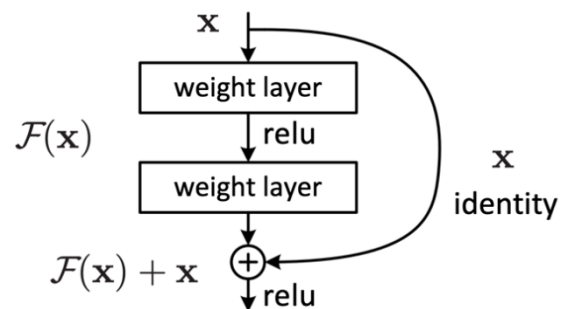


Fig. 2. a building block of residual network [40].

The presence of skip connections enables the propagation of larger gradients to initial layers. This mechanism allows the initial layers to learn at a comparable rate to the final layers, effectively addressing the issue of vanishing gradient and facilitating the training of deeper networks [42].

#### D. The Proposed Method

Detecting intrusions in IoT networks has become an increasingly formidable task due to the evolving landscape of technology. Using the unique characteristics of networked devices to detect intrusions and stop possible assaults is a major challenge. The NSL-KDD dataset has been helpful in addressing this issue by offering a wide range of features for intrusion detection. But with 41 input features, the dataset is highly dimensional and requires a feature selection approach that works. To enhance the performance of classification, reduce processing workload and prevent over fitting, this approach is essential. Identifying properties that contribute significantly to the classification process while eliminating features which are not necessary or superfluous is a major goal of most feature selection techniques. In this approach, it is essential to find features that are low in similarity with each other and have a high degree of relevance for the target variable. Conventional metrics frequently fail to account for nonlinear connections. The Chatterjee correlation coefficient, which was introduced recently, provides a new method by identifying nonlinear correlations between two vectors. It differs from traditional measures such as the Pearson correlation because of this. Fig. 3 illustrates the comparative advantage of the Chatterjee correlation, especially in capturing intricate relationships. In light of these considerations, this paper proposes the Chatterjee correlation coefficient as a robust measure of similarity between pairs of features and dissimilarity between features and targets. This contribution aids in the selection of feature subsets that optimally represent the data's characteristics.

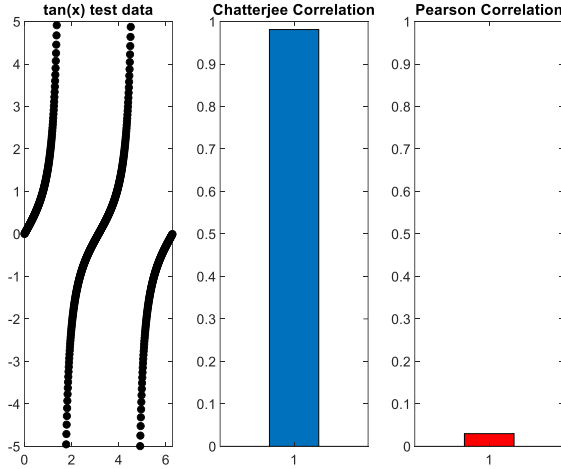


Fig. 3. Comparing the Chatterjee and Pearson correlation coefficients in capturing nonlinear correlation of function  $\tan(x)$ .

The utilization of the Chatterjee correlation as a feature subset evaluation metric underscores its role as an assessment tool rather than an algorithm for subset selection. To facilitate the dynamic selection of feature subsets, an algorithm is imperative, and in this context, metaheuristic optimization algorithms emerge as a promising choice due to their swift convergence and minimal investigations. Among these algorithms, ABC algorithm stands out as a metaheuristic

optimization technique renowned for efficiently navigating intricate solution spaces and converging to global optima.

The proposed feature selection process integrates the ABC algorithm, leveraging its capacity to explore solution spaces effectively. The Chatterjee correlation coefficient becomes an integral component of the cost function, aligning with the equations introduced in subsection A of section III. A correlation matrix is initiated to facilitate this integration, as illustrated below:

$$\forall i \in [1, n], \forall j \in [1, n]$$

$$CM_{i,j} = \begin{cases} 1/\xi_m(F_i, TA) & i = j \\ |\xi_m(F_i, F_j)| & i \neq j \end{cases} \Rightarrow \quad (11)$$

$$CM = \begin{bmatrix} 1/\xi_m(F_1, TA) & |\xi_m(F_1, F_2)| & \cdots & |\xi_m(F_1, F_n)| \\ |\xi_m(F_2, F_1)| & 1/\xi_m(F_2, TA) & \cdots & |\xi_m(F_2, F_n)| \\ \vdots & \vdots & \ddots & \vdots \\ |\xi_m(F_n, F_1)| & |\xi_m(F_n, F_2)| & \cdots & 1/\xi_m(F_n, TA) \end{bmatrix} \quad (12)$$

where  $CM$  is the proposed correlation matrix, The selected subset's number of characteristics is  $n$ ,  $F_i$  is the  $i^{\text{th}}$  feature vector,  $F_j$  is the  $j^{\text{th}}$  feature vector,  $m$  is the length of the feature vectors,  $TA$  is the target vector, and  $\xi$  is Chatterjee correlation coefficient.

By calculating the reverse Chatterjee correlation coefficient in the elements on the oblique of the confusion matrix, the level of correlation between the selected features and the target variable is assessed. Features exhibiting low absolute Chatterjee correlation values indicate significant differences between their distributions and the target, with lower diagonal values indicating higher similarity between feature and target distributions—an objective in feature selection.

This principle extends to other elements of the matrix, where lower values for non-diagonal elements suggest greater dissimilarity between each pair of selected features, contributing to the reduction of redundant information. Consequently, a lower value for each element of the  $CM$  matrix signifies an improved situation for feature selection. Moreover, considering the asymmetry of the Chatterjee correlation ( $\xi(F_i, F_j) \neq \xi(F_j, F_i)$ ), the  $CM$  is strategically designed to incorporate both terms, ensuring a comprehensive representation of the feature relationships.

Next, the summation of all elements of the matrix can be calculated in order to obtain the final cost value. However, to address the potential bias favoring smaller subsets in the cost calculation due to the summation of all matrix elements, a normalization approach is adopted to ensure a fair evaluation of subsets. This adjustment is crucial as smaller subsets inherently possess fewer elements, potentially leading to consistently lower cost values. In light of this, the normalized cost function for the ABC algorithm is formulated as follows:



$$Cost = \frac{\sum_{i=1}^n \sum_{j=1}^n CM_{i,j}}{n^2} \quad (13)$$

where *Cost* is the proposed Chatterjee-correlation-based cost function. This normalization process ensures that the cost is not unduly influenced by the subset size, providing an equitable basis for the ABC algorithm to assess and select feature subsets. The objective is to promote a balanced evaluation that considers both the quality of the features and their interrelationships, ultimately contributing to the intrusion detection method's efficacy. Following the formulation of the normalized cost function, the decision variables for feature selection are defined as binary flags, serving as indicators for whether a corresponding feature should be retained (assigned a value of 1) or discarded (assigned a value of 0). With the goal of choosing a subset of characteristics that minimizes the cost function, the optimization process iteratively and methodically modifies these choice variables.

Afterwards, in order to take use of the ResNet network's ability to capture complex correlations between features, an efficient way to translate the 1D feature vector into 2D images is developed. To perform this transformation, a diagonal matrix of size  $n \times n$  must be created, with all of the specified characteristics housed in the major diagonal. At the same time, in order to maintain the uniformity of the spacing between the features and to ensure that filters can catch all of them, these features are arranged in a different sequence by the secondary diagonal.

To get a normalized measure of how far apart the features are, we cut the feature vector in two equal parts and swap their places along the opposite diagonal. This methodical process, as shown in Figure 4, is capable of producing uniformly dispersed features distances which improve the ResNet network's ability to identify complex relationships among them.

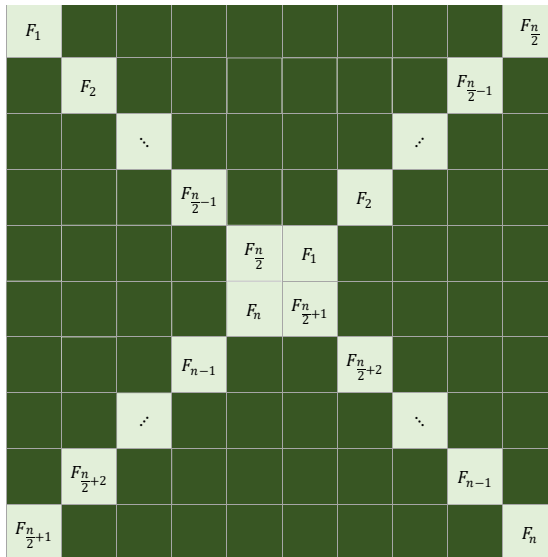


Fig. 4. The method for converting 1D features space to a 2D feature matrix is proposed.

The final stage in the proposed methodology involves the utilization of a tailored residual neural network (ResNet) for the classification process subsequent to the preprocessing of input images. This specific ResNet is crafted to alleviate the computational load associated with conventional residual neural networks, all while preserving their capacity to depict complex relationships.

The intended architectural design comprises three sets of residual blocks, each strategically contributing to the model's capacity to comprehend intricate patterns and hierarchical representations. These stacks, containing 4, 3, and 2 residual blocks respectively, empower the model to grasp both low-level and high-level features, enhancing its overall understanding of the input data. The arrangement of 16, 32, and 64 filters within each block enhances the process of feature extraction. Gradual adjustments to these filters endow the network with the capability to extract progressively complex and abstract properties, facilitating the identification of subtle details embedded in intrusion patterns.

Employing a stride of 1, a filter size of 3, and an initial filter count of 16, the initial convolutional layers play a pivotal role in shaping the learning process. This configuration adeptly extracts relevant features from the data while preserving spatial information in the input. The rationale behind these architectural decisions is rooted in the complexities of ID. The strategic placement of residual blocks and the selection of filter quantities empower the model to discern both global and local properties, facilitating a comprehensive understanding of the input data.

As the initial convolutional layers are finely tuned to detect even subtle variations, the network has the potential to promptly extract meaningful representations. The overarching objective of this ResNet configuration is to proficiently differentiate between routine and intrusive network activities through the application of hierarchical learning capabilities. The network's ability to extract and abstract features is enhanced by the deliberate placement of residual blocks and the progressive augmentation of filter count, culminating in an effective intrusion detection tool. The configuration of the planned network is presented graphically in Fig. 5. Additionally, the flowchart for the suggested solution using the aforementioned stages is shown in Fig. 6.

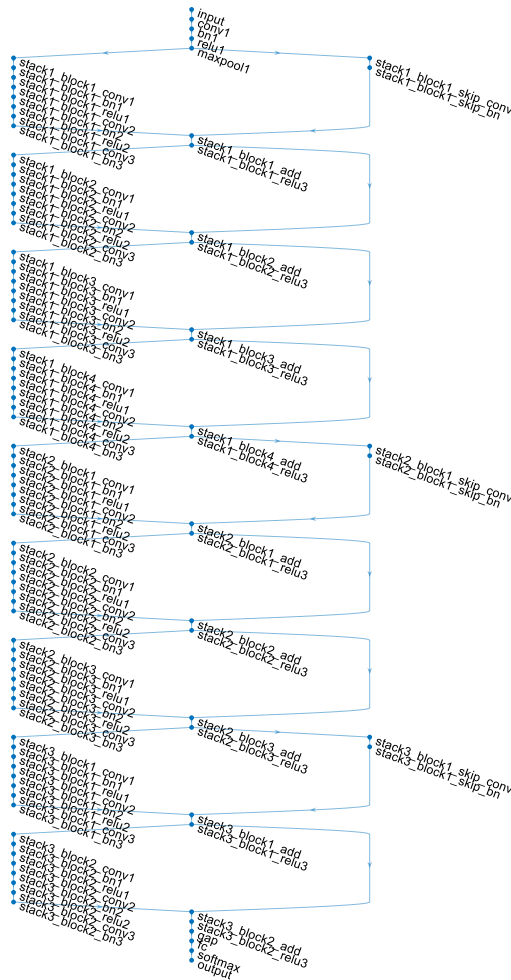


Fig. 5. The designed ResNet architecture.

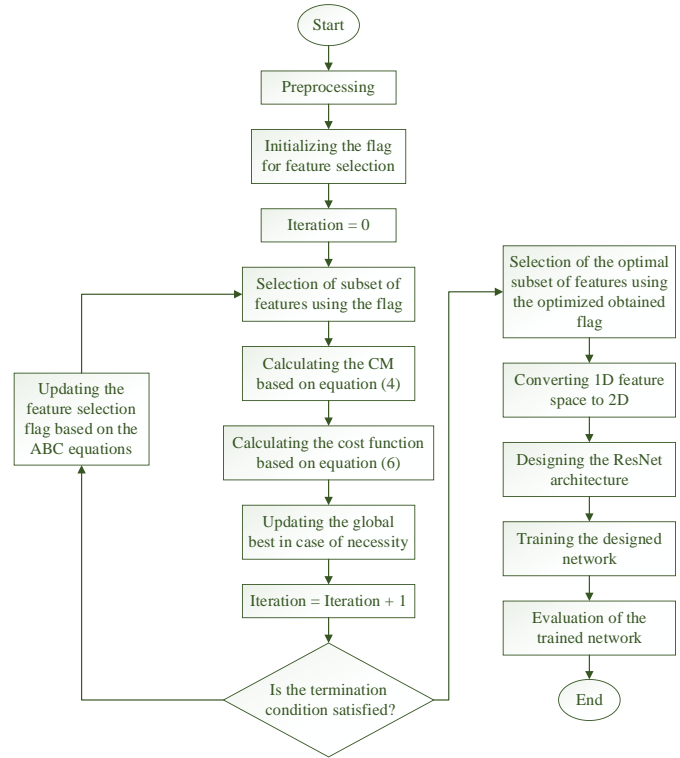


Fig. 6. The flowchart of the proposed method.

#### IV. EVALUATION METRICS

Evaluation metrics are vital in assessing the performance of intrusion detection systems (IDS). These metrics help quantify the ability of IDS to accurately detect and categorize network intrusions.

##### 1. Accuracy:

The **ratio** of accurately classified **cases** to the overall **cases** is used to determine accuracy, which is a measure of overall correctness.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

##### 2. Precision:

This **parameter** is articulated as the quotient of true positives over the entirety of instances identified as positive, serving as a gauge for the correctness of positive predictions.

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

##### 3. Recall:

Recall evaluates the capability to recognize every positive occurrence and is computed as the **ratio** of true positives relative to the entire factual positive instances.

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

##### 4. F1 Score:

This **parameter**, derived through the harmonic mean, furnishes a comprehensive assessment by finding a middle ground between recall and precision.

$$F_1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (17)$$

In these above equations:

- True Positives (TP): Items rightly recognized as positive.
- True Negatives (TN): Items rightly recognized as negative.
- False Positives (FP): Items inaccurately categorized as positive.
- False Negatives (FN): Items inaccurately categorized as negative.

Striking the right balance between precision and recall holds supreme significance for IDS. Given that the F1 score encapsulates a mean of these criterias, it delivers a holistic assessment, especially in scenarios where the frequency of intrusion instances diverges from regular occurrences. The specific priorities and requirements of the security context in question dictate the most suitable metric.

## V. SIMULATION RESULTS

This part supply a thorough elucidation of the methodology employed for assessing the proposed intrusion detection approach.

### A. Feature Selection

A crucial phase in our intrusion detection process involves feature selection, employing the ABC technique along with the Chatterjee correlation coefficient. An exhaustive assessment of the simulation outcomes, with specific attention given to the convergence behavior of the ABC algorithm and the resulting heatmap illustrating the Chatterjee correlation coefficients of the selected features.

The initial significant step in this procedure entails adjusting the variables of the ABC algorithm. In our case, careful consideration led us to determine the highest iteration count and the size of the population as 12 and 4, respectively. This adjustment stems from a meticulous weighing of the trade-offs between execution time and evaluation points. Although it demands increased computational time, a more thorough evaluation may yield a more precise optimization. Conversely, a larger population size may extend execution times but allows for a broader exploration of the solution space. Our selected settings strive for efficient optimization without unreasonably increasing the execution duration. The convergence curve of the ABC method across iterations is illustrated in Fig. 7. The quantity of function evaluations is depicted along the x-axis, whereas the convergence behavior is shown along the y-axis. The graph shows how well the algorithm refines the feature subset in each iteration. The parameters used enable the ABC method to converge within the given restrictions while maintaining a balance between computing efficiency and optimization accuracy.

After feature selection process, 24 features are selected as final optimal features. Fig. 8 showcases the resulting heatmap of the Chatterjee correlation coefficients for the selected features. The correlation strength between a pair of features is shown by each cell in the heatmap. yellow colors signify higher correlation, while green colors indicate lower correlation.

Contrary to the previous analysis, the heatmap not only illustrates the low correlation between each pair of selected features but also signifies the removal of redundant features. This outcome indicates a remarkable ability to extract unique

characteristics from each selected feature, highlighting the effectiveness of our methodology in eliminating redundancy. The low correlation emphasizes that the selected features possess distinct attributes, showcasing their individual relevance for intrusion detection.

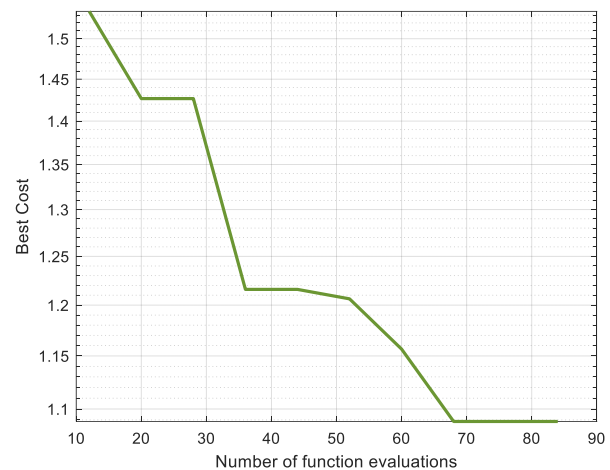


Fig. 7. The convergence curve of ABC optimization algorithm for feature selection.

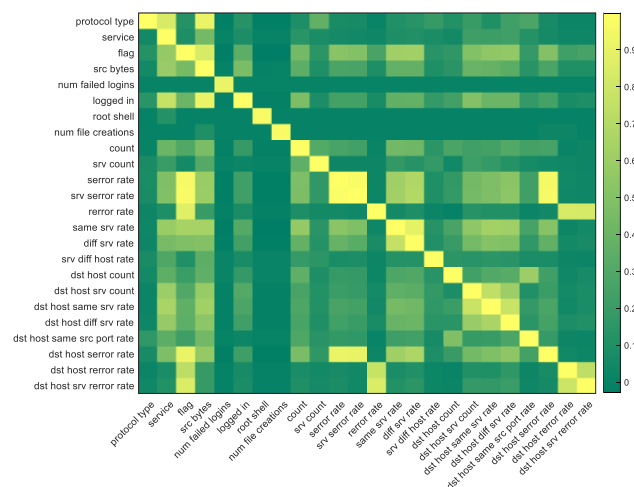


Fig. 8. The selected features' Chatterjee correlation heatmap.

Finally, a maximum of 100 epochs is chosen, accompanied by a batch size of 16. These selections strive to achieve equilibrium between the speed of convergence and the accuracy of classification. The utilization of the ADAM optimization algorithm is preferred for its swift convergence and other advantageous properties that enhance the efficiency of model training. Fig. 9 visually illustrates the convergence of the loss function over epochs, providing insights into the optimization process.

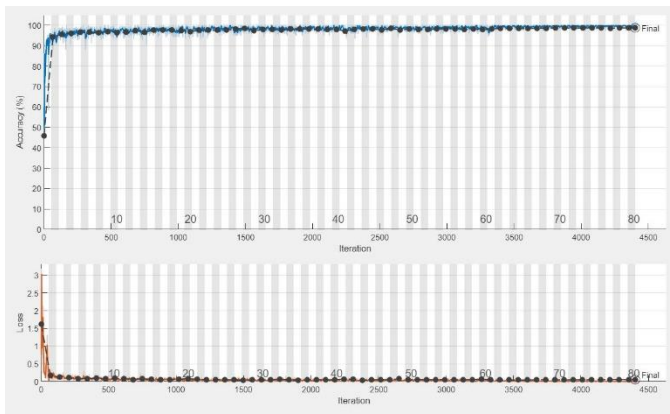


Fig. 9. The optimization progress of the designed ResNet.

### B. Confusion Matrix

The proposed intrusion detection method's effectiveness is evaluated using confusion matrices, offering a detailed breakdown of classification outcomes into true positives (accurate intrusion identifications), true negatives (correct identification of normal behavior), false positives (normal instances mistakenly classified as intrusions), and false negatives (intrusions wrongly classified as normal instances). Fig. 10 and Fig. 11 illustrate these matrices for the training and testing phases, respectively. This thorough analysis of confusion matrices lays the groundwork for a more in-depth exploration of ROC curves and performance criteria, providing an extensive assessment of the intrusion detection method's efficacy across a diverse range of network intrusion scenarios. Analyzing the training matrix in Fig. 10 reveals strong diagonal elements, indicating precise classification for each class; however, attention to off-diagonal elements is crucial for understanding misclassifications. Moreover, the evaluation matrix depicted in Figure 11 illustrates the model's capacity to apply acquired knowledge to novel data, offering valuable insights into potential challenges and opportunities for enhancement in specific categories.



Fig. 10. Confusion Matrix for Training Data.

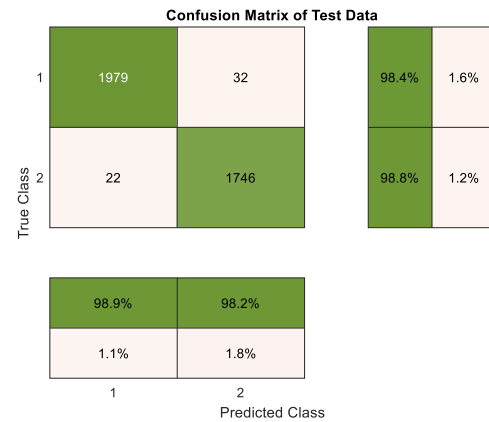


Fig. 11. Confusion Matrix for Test Data.

### C. Receiver Operating Characteristic

This curve serves as a means to analyze the effectiveness of the suggested ID technique in discerning between benign and intrusive instances. These curves offer an appraisal of the model's discriminatory prowess by delineating the balance between true positive rates and false positive rates across divergent classification boundaries. Figures 12 and 13 display the ROC curves for the testing and training phases, respectively, showcasing optimal performance. These curves shed light on the model's overall robustness by revealing its capability to differentiate between various types of intrusions in both phases. A meticulous examination of the ROC curves allows for an in-depth analysis of the model's efficacy at different decision benchmarks. This scrutiny furnishes a nuanced understanding of the method's proficiency to strike a balance between specificity and sensitivity, crucial for effective intrusion detection. Further exploration of the performance criteria in subsequent sections helps with a comprehensive grasp of how well the proposed approach operates in diverse intrusion scenarios.

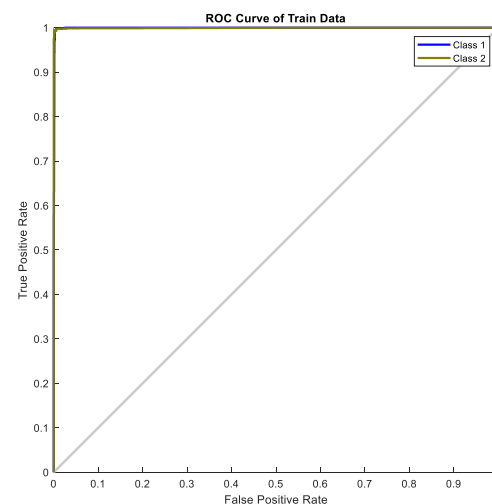


Fig. 12. ROC Curve for Training Data.



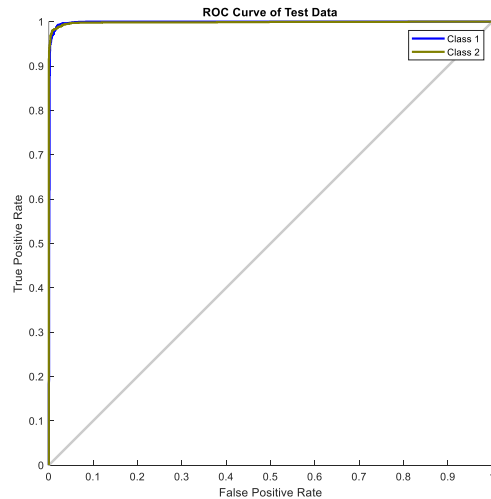


Fig. 13. ROC Curve for Test Data.

#### D. Evaluation Performance Utilizing Assessment Metrics

The outcomes of the ultimate intrusion detection evaluation are illustrated in Figures 14 and 15. These visuals depict a noteworthy enhancement in intrusion detection, showcasing precision rates of 99.727% and 98.7557%, along with accuracy rates of 99.535% and 98.5711% for the train and test phases, in sequence. These outcomes demonstrate the exceptional proficiency of the ResNet in detecting intrusions and the usefulness of feature selection using the Chatterjee and Artificial Bee Colony algorithms. The substantial increase in intrusion detection accuracy demonstrates the network's effectiveness not only in thwarting attacks but also in effectively managing challenges and adapting to dynamic threats within the network environment.

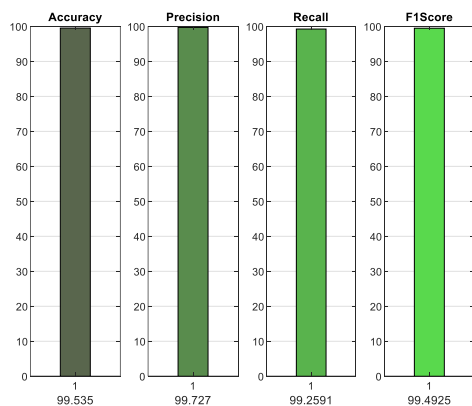


Fig. 14. The evaluation of modelling using evaluation metrics for train dataset.

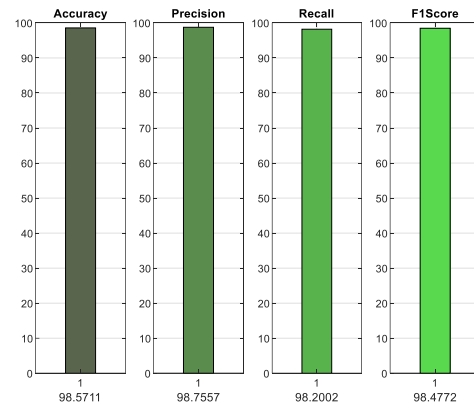


Fig. 15. The evaluation of modelling using evaluation metrics for test datasets.

## VI. COMPARISON

In order to evaluate how well the suggested method with the existing literature, four recently published studies are employed. ML algorithms, including DT, RF, and XGBoost, are employed in article [43] for NIDS within the SDN controller to detect vicious behavior in network traffic, utilizing the NSL-KDD. Advanced preprocessing techniques are applied to enhance data quality, achieving outstanding results with a 95.95% accuracy in detection and classification of (DDoS, PROBE, R2L, U2R) attacks employing exclusively 5 out of 41 features of NSL-KDD.

In [44], a DL model for network ID (DLNID) is put forth. It incorporates a Bi-LSTM network and an attention mechanism. Adaptive synthetic sampling (ADASYN) is used to address data imbalance, and a modified stacked autoencoder is used to reduce dimensionality. DLNID reaches greater accuracy 90.73% and F1 score 89.65% when tested on the NSL-KDD.

A PSO-LightGBM approach is presented in [45]. This method is employed to extract data features, which are then channeled into a one-class SVM. The UNSW-NB15 is employed to confirm the suggested method. Experimental findings show accurate detecting in a variety of harmful data, including tiny sample data like worms, shellcode, and backdoors, with an accuracy of 86.68%.

In reference [46], a description is provided for an intrusion detection system (IDS) using the Soft-Roof-Sign activation function based on the Tree-CNN. The system demonstrates an average detection accuracy of 98% across various attack types, including DDoS, Infiltration, Brute Force, and Web attacks. An evaluation of the effectiveness within a medium-sized company highlights its low complexity and reduced processing time.

The summarized comparative outcomes of the recommended process against the introduced approaches are showcased in Table I. Notably, the suggested method surpasses other approaches in terms of accuracy.

TABLE I. COMPARING THE SUGGESTED APPROACH WITH OTHER LITERATURE.

Reference	Method	Dataset	Accuracy
[40]	DT, RF, and XGBoost	NSL-KDD	95.95%

[41]	DLNID (Deep Learning Model for NID) and Bi-LSTM	NSL-KDD	90.73%
[42]	PSO-LightGBM	UNSW-NB15	86.68%
[43]	Tree-CNN with SRS activation function	dataset built of the university campus, the medium-sized and the CICIDS2017	98 %
The proposed method	Chatterjee correlation coefficient-ABC optimization algorithm-ResNet	NSL-KDD	98.5711%

## VII. CONCLUSION

In overview, this study introduces an inventive and effective approach for detecting intrusions in Internet of Things (IoT) networks, employing a tailored residual neural network (ResNet), artificial bee colony optimization, and the Chatterjee correlation coefficient. The methodology, meticulously delineated through its phases, addresses key concerns related to nonlinear relationships in feature selection, enhances the ResNet's ability to capture intricate patterns, and streamlines the selection process using metaheuristic optimization.

The utilization of the Chatterjee correlation coefficient provides a nuanced perspective, particularly in handling duplicate data, showcasing its adaptability to various dataset configurations. The ABC optimization augments the efficiency of feature selection by ensuring a well-balanced subset for ResNet classification. When thoughtfully crafted, the ResNet architecture proves vital in discerning both high-level and low-level properties, rendering it a potent tool for ID.

The efficacy of the proposed methodology is evidenced by empirical findings, demonstrating its capability to detect intrusions in IoT networks with a 98.5711% accuracy on the test dataset. Comparative analysis with other approaches underscores the strengths and advantages of the Chatterjee correlation-driven method, suggesting its potential to address current gaps in intrusion detection research.

This research extends the application of the Chatterjee correlation coefficient in cybersecurity while presenting a novel approach to intrusion detection. Robust simulation results and comprehensive assessment metrics affirm the resilience of our methodology. The proposed approach opens avenues for future research, paving the way for advancements in IoT network security.

## REFERENCES

- [1] Fathollahi, L., Feghhi, M. M., & Atashbar, M. (2024). Energy optimization for full-duplex wireless-powered IoT networks using rotary-wing UAV with multiple antennas. *Computer Communications*, 215, 62-73.
- [2] C. Frank, C. Nance, S. Jarocki and W. Pauli, "Protecting IoT from Mirai botnets; IoT device hardening," in *Proceedings of the conference on information systems applied research*, Austin, Texas, 2017.
- [3] Asbaghi, S. S., & Feghhi, M. M. (2023). Resource Allocation for Secure Ultra-Reliable Low-Latency-Communication in IoT Applications. *arXiv preprint arXiv:2312.10555*.
- [4] Albulayhi, K., Smadi, A. A., Sheldon, F. T., & Abercrombie, R. K. (2021). IoT intrusion detection taxonomy, reference architecture, and analyses. *Sensors*, 21(19), 6432.
- [5] Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications*, 186, 115782.
- [6] Sajja, G. S., Mustafa, M., Ponnusamy, R., & Abdufattokhov, S. (2021). Machine learning algorithms in intrusion detection and classification. *Annals of the Romanian Society for Cell Biology*, 25(6), 12211-12219.
- [7] Alkadi, O., Moustafa, N., Turnbull, B., & Choo, K. K. R. (2020). A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. *IEEE Internet of Things Journal*, 8(12), 9463-9472.
- [8] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics*, 8(11), 1210.
- [9] Kumar, V., Sinha, D., Das, A. K., Pandey, S. C., & Goswami, R. T. (2020). An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. *Cluster Computing*, 23, 1397-1418.
- [10] Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, 111, 2287-2310.
- [11] Song, H. M., Woo, J., & Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications*, 21, 100198.
- [12] Zhang, H., Li, J. L., Liu, X. M., & Dong, C. (2021). Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. *Future Generation Computer Systems*, 122, 130-143.
- [13] Rahman, S. A., Tout, H., Talhi, C., & Mourad, A. (2020). Internet of things intrusion detection: Centralized, on-device, or federated learning?. *IEEE Network*, 34(6), 310-317.
- [14] Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., ... & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*, 110, 102448.
- [15] Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). CNN-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6), 916.
- [16] Qi, L., Yang, Y., Zhou, X., Rafique, W., & Ma, J. (2021). Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0. *IEEE Transactions on Industrial Informatics*, 18(9), 6503-6511.
- [17] Nguyen, M. T., & Kim, K. (2020). Genetic convolutional neural network for intrusion detection systems. *Future Generation Computer Systems*, 113, 418-427.
- [18] Abbas, A., Khan, M. A., Latif, S., Ajaz, M., Shah, A. A., & Ahmad, J. (2021). A new ensemble-based intrusion detection system for internet of things. *Arabian Journal for Science and Engineering*, 1-15.
- [19] Halimaa, A., & Sundarakantham, K. (2019, April). Machine learning based intrusion detection system. In *2019 3rd International conference on trends in electronics and informatics (ICOEI)* (pp. 916-920). IEEE.
- [20] Balyan, A. K., Ahuja, S., Lilhore, U. K., Sharma, S. K., Manoharan, P., Algarni, A. D., ... & Raahemifar, K. (2022). A hybrid intrusion detection model using ega-pso and improved random forest method. *Sensors*, 22(16), 5986.
- [21] Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). Intrudtree: a machine learning based cyber security intrusion detection model. *Symmetry*, 12(5), 754.
- [22] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access*, 7, 41525-41550.
- [23] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for

- detecting internet of things network attacks. *Alexandria Engineering Journal*, 61(12), 9395-9409.
- [24] Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020, September). A machine learning approach for intrusion detection system on NSL-KDD dataset. In 2020 international conference on smart electronics and communication (ICOSEC) (pp. 919-924). IEEE.
- [25] Choraś, M., & Pawlicki, M. (2021). Intrusion detection approach based on optimised artificial neural network. *Neurocomputing*, 452, 705-715.
- [26] Roy, S., Li, J., Choi, B. J., & Bai, Y. (2022). A lightweight supervised intrusion detection mechanism for IoT networks. *Future Generation Computer Systems*, 127, 276-285.
- [27] Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet of Things Journal*, 7(8), 6882-6897.
- [28] Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. *Procedia Computer Science*, 185, 239-247.
- [29] Liang, W., Xiao, L., Zhang, K., Tang, M., He, D., & Li, K. C. (2021). Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems. *IEEE Internet of Things Journal*, 9(16), 14741-14751.
- [30] Alavizadeh, H., Alavizadeh, H., & Jang-Jaccard, J. (2022). Deep Q-learning based reinforcement learning approach for network intrusion detection. *Computers*, 11(3), 41.
- [31] Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). Ieee.
- [32] Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International journal of advanced research in computer and communication engineering*, 4(6), 446-452.
- [33] Chatterjee, Sourav. "A new coefficient of correlation." *Journal of the American Statistical Association* 116, no. 536 (2021): 2009-2022.
- [34] Teodorovic, D., Lucic, P., Markovic, G., & Dell'Orco, M. (2006, September). Bee colony optimization: principles and applications. In 2006 8th seminar on neural network applications in electrical engineering (pp. 151-156). IEEE.
- [35] Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, 214(1), 108-132.
- [36] Sharma, A. B. H. I. S. H. E. K., Sharma, A. B. H. I. N. A. V., Choudhary, S. A. C. H. I., Pachauri, R. K., Shrivastava, A. A. Y. U. S. H., & Kumar, D. E. E. P. A. K. (2020). A review on artificial bee colony and it's engineering applications. *Journal of Critical Reviews*, 7(11), 4097-4107.
- [37] Bansal, J. C., Gopal, A., & Nagar, A. K. (2018). Stability analysis of artificial bee colony optimization algorithm. *Swarm and evolutionary computation*, 41, 9-19.
- [38] Hassan, A. N., Mohassel Feghhi, M., & Esmaeili, V. (2021). A Fast Automatic Modulation Classification Based on STFT Using Hybrid Deep Neural Network. *Journal of Communication Engineering*, 10(2).
- [39] Residual blocks—Building blocks of ResNet. <https://towardsdatascience.com/residual-blocksbuilding-blocks-of-resnet-fd90ca15d6ec>
- [40] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [41] Sarwinda, D., Paradisa, R. H., Bustamam, A., & Anggia, P. (2021). Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer. *Procedia Computer Science*, 179, 423-431.
- [42] Mhapsekar, M., Mhapsekar, P., Mhatre, A., & Sawant, V. (2020). Implementation of residual network (ResNet) for devanagari handwritten character recognition. In *Advanced Computing Technologies and Applications: Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA 2020* (pp. 137-148). Springer Singapore.
- [43] Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet*, 13(5), 111.
- [44] Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A deep learning model for network intrusion detection with imbalanced data. *Electronics*, 11(6), 898.
- [45] Liu, J., Yang, D., Lian, M., & Li, M. (2021). Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access*, 9, 38254-38268.
- [46] Mendonça, R. V., Teodoro, A. A., Rosa, R. L., Saadi, M., Melgarejo, D. C., Nardelli, P. H., & Rodríguez, D. Z. (2021). Intrusion detection system based on fast hierarchical deep convolutional neural network. *IEEE Access*, 9, 61024-61034.