DOI: 10.24996/ijs.2025.66.2.19

ISSN: 0067-2904

Enhancing IoT Security: An Optimization Algorithm for Fog Layer-Based DDoS Attack Mitigation Framework

Mehdi Ebady Manaa[1*,2], Fryal Jassim Abd Al-Razaq[3], Hussein A. A. Al-Khamees[4]

[1]*Department of Artificial Intelligence, College of Science, Al-Mustaqbal University, 51001, Babylon, Iraq*
[2]*Department of Information Technology, Information Technology College, University of Babylon, Babylon, Iraq*
[3]*Department of Software, College of Information Technology, University of Babylon, Babylon, Iraq;*
[4]*Department of Computer Techniques Engineering, Engineering, College of Engineering and Engineering Techniques, Al-Mustaqbal University, 51001, Babylon, Iraq*

Received: 11/10/2023 Accepted: 26/1/2024 Published: 28/2/2025

Abstract

The Internet of Things (IoT) refers to a network comprised of interconnected items, including computing devices and digital gadgets. Cloud-based IoT infrastructures are vulnerable to distributed denial of service (DDoS) attacks. A DDoS attack has the potential to incapacitate a server for an extended duration, resulting in service disruptions as a consequence of overwhelming system resources. This research presents a novel framework for mitigating DDoS attacks in IoT networks. The proposed system leverages the fog-cloud architecture to provide efficient, lightweight, and precise attack mitigation. Notably, the mitigation process is executed at the fog layer. The suggested fog layer uses Particle Swarm Optimization (PSO) to make allocating resources easier, which makes it possible for the mitigation framework to be set up quickly. This approach addresses the challenges associated with resource management on resource-constrained IoT devices. The mitigation framework uses the Fitness Leader Optimization (FLO) approach to construct a trained database, taking into consideration factors such as the needed time, the size of the request, and the number of created requests. The FLO system employs multilayer perceptron (MLP), k-nearest neighbors (KNN), and support vector machine (SVM) classification algorithms to effectively mitigate the assault. The results of this study show that adding classification algorithms to our framework made it easier to test networks for Internet of Things (IoT) devices, especially when the Particle Swarm Optimization (PSO) method was used together. The mitigation framework demonstrates a minimized fitness value of 0.284556 seconds, showcasing enhanced resource utilization and processing time optimization for IoT nodes and servers in a distributed fog environment. The total average of resource utilization is improved to 6.0850%, processing time is decreased to 17.0397, and fitness value is decreased to 0.0258 seconds in the proposed DDoS attack mitigation system. The machine learning classification model achieves high accuracy, with SVM leading at 99.6785% compared to others, emphasizing the robustness of the proposed framework in securing IoT networks.

Keywords: Anomaly Detection, Internet of Things (IoT), DDoS mitigation, Fog Computing, Classification Algorithms.

* Email: mahdi.ebadi@uomus.edu.iq

تعزيز أمن إنترنت الأشياء: خوارزمية تحسين لإطار عمل تخفيف هجمات الحرمان من الخدمة الموزع المستندة إلى طبقة الضباب

مهدي عبادي مانع[1,2], فريال جاسم عبد الرزاق [3], حسين عبد الأمير عباس الخميس[4]

[1]قسم الذكاء الاصطناعي , كلية العلوم, جامعة المستقبل,51001 بابل, العراق

[2] قسم شبكات المعلومات، كلية تكنلوجيا المعلومات , جامعة بابل, بابل, العراق

[3] قسم البرمجيات، كلية تكنولوجيا المعلومات، جامعة بابل، بابل، العراق

[4] قسم هندسة تقنيات الحاسوب, كلية الهندسة والتقنيات الهندسية, , جامعة المستقبل,51001 بابل, العراق

الخلاصة

يشير إنترنت الأشياء (IoTs) إلى شبكة تتكون من عناصر مترابطة، بما في ذلك أجهزة الحاسوب والأدوات الرقمية. البنى التحتية لإنترنت الأشياء المستندة إلى السحابة معرضة لهجمات الحرمان من الخدمة الموزعة (DDoS). من المحتمل أن يؤدي هجوم DDoS إلى تعطيل الخادم لفترة طويلة، ويؤدي إلى انقطاع الخدمة نتيجة لاستنزاف موارد النظام. يقدم هذا البحث إطارًا جديدًا للتخفيف من هجمات DDoS في شبكات إنترنت الأشياء. يعمل النظام المقترح على تعزيز بنية سحابة الضباب لتوفير منع فعال وخفيف الوزن ودقيق للهجوم. والجدير بالذكر أن عملية منع الهجوم يتم تنفيذها في طبقة الضباب. تشتمل طبقة الضباب المقترحة على تحسين سرب الجسيمات (PSO) لتسهيل تخصيص الموارد، وبالتالي تمكين النشر الفعال لإطار منع الهجوم. يعالج هذا النهج التحديات المرتبطة بإدارة الموارد في أجهزة إنترنت الأشياء المحدودة الموارد. يستعمل إطار منع الهجوم نهج تحسين اللياقة البدنية (FLO) لإنشاء قاعدة بيانات مدربة، مع الأخذ في الاعتبار عوامل مثل الوقت المطلوب وحجم الطلب وعدد الطلبات التي تم إنشاؤها. يستعمل نظام FLO خوارزميات تصنيف الإدراك الحسي متعدد الطبقات (MLP) و(KNN) k-Nearest Neighbors وآلة دعم المتجهات (SVM) للتخفيف ومنع الهجوم بشكل فعال. تشير نتائج هذه البحث إلى أن تنفيذ خوارزميات التصنيف داخل إطار عملنا أدى إلى تحسين نتائج تقييم الشبكة لأجهزة إنترنت الأشياء، خاصة عند دمجها مع طريقة تحسين سرب الجسيمات (PSO). يظهر إطار التخفيف قيمة لياقة مُقللة تبلغ 0.284556 ثانية، مما يُظهر تحسنًا في استعمال الموارد وتحسين وقت المعالجة لأجهزة انترنيت الاشياء والخوادم في بيئة الضباب الموزعة. حيث يتم تحسين إجمالي متوسط استعمال الموارد إلى 6.0850٪، وتم تقليل وقت المعالجة إلى 17.0397، وتقليل قيمة اللياقة إلى 0.0258 ثانية لنظام التخفيف من هجمات DDoS في النظام المقترح. يحقق نموذج التصنيف باستعمال تعلم الآلة دقة عالية، حيث يتصدر SVM بنسبة 99.6785٪ مقارنة بالنماذج الأخرى، مما يبرز قوة الإطار المقترح في تأمين شبكات انترنت الاشياء..

1. Introduction

The IoT refers to a network of intelligent devices that are deployed in the physical world and broadcast their data in real-time over the internet [1]. The primary objective of IoT devices is to quantify physical attributes and communicate this data in digital format to a cloud server, where it undergoes processing [2]. The IoT network is comprised of many components, including sensors, a wireless network infrastructure, a data storage device, a computer platform, and a user application [3]. In the context of an IoT network, smart devices engage in the exchange of information and thereafter execute their designated tasks autonomously. For instance, in a smart home environment, various smart gadgets, such as an electric toaster or coffee maker, operate without the need for human intervention [4].

The lack of a defined structure in the IoT renders it susceptible to numerous DDoS threats [5]. The amalgamation of fog computing with the IoT has established a proficient foundation for the execution of an anomaly mitigation approach specifically aimed at resolving security concerns like DDoS attacks [6]. Fog computing enhances the security paradigm by providing an elevated level of protection and implementing several layers of defense [7]. The integration of machine learning techniques into fog computing systems has the potential to enhance network routing performance and offer more comprehensive insights and solutions [8]. The platform under consideration offers a virtualized environment that enables the delivery of computational, storage, and networking services to allow data interchange between end-user devices and data centers. Similar to cloud computing, the concept of fog computing relies on the presence and accessibility of computational, storage, and networking resources [9].

In order to mitigate the challenges associated with cloud computing, it is imperative that these resources be situated in close physical proximity to consumers [10]. Fog nodes can manifest as either servers or networking equipment that has supplementary processing capabilities. Wireless access points have the potential to incorporate them as well. Fog nodes are commonly situated in the periphery of the network, close to end users of the IoT [11].

This paper presents a suggested framework for mitigating DDoS attacks. The utilization of fog computing is employed to implement an anomaly mitigation framework that is especially tailored for IoT networks. This can be achieved through the utilization of a resource allocation optimization algorithm and machine learning techniques. In contrast to the currently used approaches, our model offers rapid detection of DDoS attacks, exhibiting superior accuracy and little occurrence of false positives, while also demonstrating scalability. The framework, after undergoing training, is capable of categorizing network traffic as either legitimate or illegitimate. This categorization occurs prior to the traffic being transmitted to the IoT gateway basebroker and fog node. The database stores the signature of any detected unauthorized traffic. This storage serves the purpose of facilitating computation and ensuring a quicker response in the event that the identical form of assault is once again carried out.

The remaining sections of this paper are structured as follows: Section (2) is an overview of the commonly related works for different studies for securing the Internet-of-Things (IoT) fog layer against DDoS attacks. Section (3) explains the main steps to implement the proposed system. Section (4) describes the modeling configuration and results, and finally, Section (5) illustrates the paper conclusion.

2. Related Work

This section presents an overview of several DDoS mitigation strategies within the context of IoT systems utilizing a fog computing network architecture. Several of the proposed systems will be discussed in detail below.

In a previous study [12], the authors introduced two artificial intelligence approaches, namely Random Forest (RF) and XGBoost. These techniques have been included in a security framework, providing comprehensive autonomy in decision-making skills. Furthermore, the utilization of the Interplanetary File System (IPFS) is proposed as a solution for achieving data load balancing and distributed file storage of IoT data. The employed approach presents a decentralized framework that relies on fog computing to identify DDoS threats in smart contracts. The evaluation of the detection system is conducted using a real-world dataset specific to the IoT, known as BoT-IoT. The system under consideration is assessed based on its accuracy (AC), detection rate (DR), and false alarm rate (FAR). The findings validate the efficacy of the suggested framework in comparison to many contemporary state-of-the-art methodologies for identifying infrequent assaults. The suggested framework succeeded in obtaining a DR of 99.99% through the utilization of the RF algorithm, employing a set of 10 characteristics extracted from the BoT-IoT dataset.

The architecture of an IoT-Fog layer has been implemented with adequate computational capabilities above the end device layer, as described in [13]. Two deep-learning-based models have been utilized. The initial approach involves utilizing a Long Short-Term Memory (LSTM) model to discern between harmful and benign data. Subsequently, a Convolutional Neural Network (CNN) model is employed to further categorize the data into several attack types. The LSTM model utilized in this study demonstrates a classification accuracy of 98%, whereas the CNN model exhibits an accuracy of 86%.

In [14], a framework was presented for the categorization of intrusions, employing a fusion of CNN and LSTM. This methodology leverages the capabilities of deep learning methodologies to attain accurate forecasts of such attacks. It also uses the CNN with the LSTM-based Fog Computing Intrusion Detection (ICNN-FCID) model to sort different kinds of attacks into different groups. The used model is demonstrated using NSL-KDD, a well-established benchmark dataset, and demonstrates a noteworthy level of accuracy in detecting attacks, reaching 96.5%. Comparative analyses were undertaken to assess the efficacy of this model in comparison to traditional and modern deep learning approaches. The results indicate the model demonstrated superior performance. The ICNN-FCID model demonstrates its suitability for use in fog layer devices since it effectively enables the surveillance of network traffic and the identification of possible security breaches. As a result, it is possible to ensure the protection of cloud servers and fog layer devices against malicious users, hence maintaining their continual accessibility for the provision of services to IoT devices.

The authors of [15] provided a novel approach referred to as the Deep Intelligent DDoS Attack Detection Method (DI-ADS). The primary objective of DI-ADS is to effectively identify and mitigate DDoS assaults inside fog-based IoT systems. The system primarily relies on the deployment of a DLM to identify and mitigate DDoS attacks inside the network. The implementation of the DLM occurs in the computational model of the fog node. This model is tasked with anticipating the behavior of the IoT device at the end. To ascertain the most suitable DLM model for the fog layer, a comprehensive performance evaluation is carried out on several models, namely DNMLP, LSTM, SVM, KNN, LR, and RF. The simulation is run on the Python Anaconda framework using a new DDoS-SDN dataset that was obtained from Mendeley. This dataset includes three types of DDoS attacks: TCP Syn-Flood, UDP Flood, and ICMP attacks. The results indicate that the DNMLP model had superior accuracy since it achieved a rate of 99.44% in comparison to other DL and ML models. The proposed methodology examines DNMLP as a viable choice for deployment at the fog layer owing to its exceptional efficacy in detecting DDoS attacks when compared to conventional approaches.

According to [16], it conducted an analysis of the risks and assaults that occur in the Industrial Internet of Things (IIoT) framework. They also explore the effects of DDoS attacks on the production process and the resulting communication dysfunctions that arise within IIoT services and applications. This article furthermore presented a proposed reference security architecture that amplifies the benefits of fog computing in order to showcase countermeasures against DDoS assaults as well as potential ways for effectively mitigating such attacks on a large scale. The results showed an increase in the efficiency of device authorization and data responsibility speed, but they did not work on accuracy.

In a study [17], the authors have devised a highly effective system for mitigating anomalies in the IoT network. This system involves the creation and deployment of a DDoS attack detection mechanism that employs a statistical approach, integrating three distinct algorithms: EWMA, KNN, and the CUSUM. The amalgamation of fog computing and the IoT has engendered a robust framework for the deployment of an anomaly mitigation approach, which serves as a viable solution to tackle security concerns, including the menace posed by botnet attacks. The module under consideration was assessed using the Bot-IoT dataset. This system had an overall accuracy of 99% while maintaining a minimal FPR. Furthermore, the researchers have successfully obtained favorable outcomes in the differentiation of IoT devices from non-IoT gadgets. This accomplishment will be beneficial for networking teams as they endeavor to discern between these two categories.

In [18], it is explained how to use a CNN model along with a bidirectional gated recurrent unit (Bi-GRU) model to find and classify intruders. The BiGRU model incorporates an attention method to discover the crucial aspects that contribute to the detection of DDoS attacks. Additionally, the use of the Wild Horse Optimization (WHO) algorithm, a nature-inspired meta-heuristic optimization technique, improves the classification model's precision. The system that has been provided demonstrates superior performance compared to the already available approaches in terms of accuracy reaching 99.35%, detection rate of 98.99%, precision of 99.9%, and F1-Score of 99% when applied to the APA DDoS attack dataset. Moreover, it achieves a high level of accuracy of 99.7%, a detection rate of 99%, precision of 99.89%, and an F1-score of 99% when applied to the ToN-IoT dataset.

The authors of [19] utilized a hybrid feature selection scheme that combines statistical test-based filter approaches, including Chi-Square (XX^2), Pearson’s Correlation Coefficient (PCC), and Mutual Information (MI), with a metaheuristic approach called Non- Dominated Sorting Genetic Algorithm (NSGA-II) for the purpose of optimizing features. The suggested approach utilizes filter-based techniques to prioritize the features for guided population initialization in NSGA-II, resulting in an expedited convergence towards a solution. The performance evaluation of the suggested method is conducted by utilizing the ToN-IoT dataset, with a focus on two key metrics: the number of chosen features and the accuracy achieved. The experimental results are contrasted with contemporary state-of-the-art approaches. The examination of the results reveals the exceptional performance of the suggested scheme, which utilizes a minimal amount of optimized features (specifically, just 13 out of the total 43 characteristics). Moreover, a good accuracy of 99.48% was achieved by this scheme.

In the work of [20], the authors introduced a feature selection algorithm based on the Firefly Algorithm (FA) to enhance IDS performance. Additionally, the Naïve Bayesian classifier is employed to distinguish between attacks and normal behaviors in network traffic. The FA algorithm is utilized to select discriminative features from the NSL-KDD dataset. The NSL-KDD dataset consists of main attacks such as DoS, User to Root (U2R), Remote to Local (R2L), and probing. The results achieved and accuracy exceeded 94.83% with the number of features (15) using FA and NB algorithms.

In the context of securing data communication across networks, the use of network intrusion detection systems (IDS) has proven to be crucial for a comprehensive defense against intrusion and abuse [21]. Traditionally, research and applications in intrusion detection systems have relied on analyzing datasets containing various attack types using batch learning machine classification. One specific attack addressed in this work is the DDoS attack, which is characterized by overwhelming victim resources and bandwidth. However, the presented work introduces an intrusion detection system based on data stream classification, applying several data stream algorithms to the CICIDS2017 datasets. The evaluation of results considers both high accuracy (more than 98%) and low computation time for the SGD and OzaBoost algorithms.

A framework to enhance security through the multi-class classification of IoT botnet attacks (IBA) using a high-dimensional dataset was proposed by [22]. The dimensionality reduction technique engaged is a classifier-based feature selection with an extra tree classifier (EXT), and the performance is compared with other techniques such as random forest classifiers and principal component analysis. The results show that KNN achieved an accuracy exceeding 99.5% across all datasets when utilizing principal components (PC) ranging from 64 to 10.

3. The Proposed System

The proposed system concerns the architectural design of a distributed resource allocation system for IoTs devices, which utilizes a Fog-based approach. The system comprises three distinct layers. Each of them is simulated and created with the C++ code design and simulated with OMNET++ and machine learning applied to Java programming language. The first layer is the IoTs layer, which consists of wireless IoT nodes to transmit / received data, access point to provide a coverage area for wireless connection, router to provide routing purposes for arrived packets with base-broker as virtualize layer to provide computing services controlled by fog node as the determination of the priority of jobs to be executed first is a crucial aspect of task management.

The second layer comprises a distributed Fog resource allocation that operates on specific Fog nodes. This layer is responsible for controlling the network components, allocating resources, and mitigating DDoS attacks through network transmission. Furthermore, it facilitated optimal decision-making for time-sensitive requests (tasks) originating from the device layer, taking into account the large number of requesters, technical and quality-of-service requirements of customers, as well as the scope and constraints of server provider resources. The PSO optimizers utilized in fog nodes exhibit rapid adaptability to the allocation and release of resources based on requests. The third layer of cloud servers functions as network service components that furnish responses to individual requests made by IoT devices. Furthermore, it provides an Intrusion Response, Storage, Communication, and Statistical Module; it also facilitates the consolidation of resources for the purpose of response processing and storage provision. Figure 1 depicts a fog computing system that utilizes a PSO optimization algorithm to achieve resource allocation in the context of IoT fog-cloud server networks.

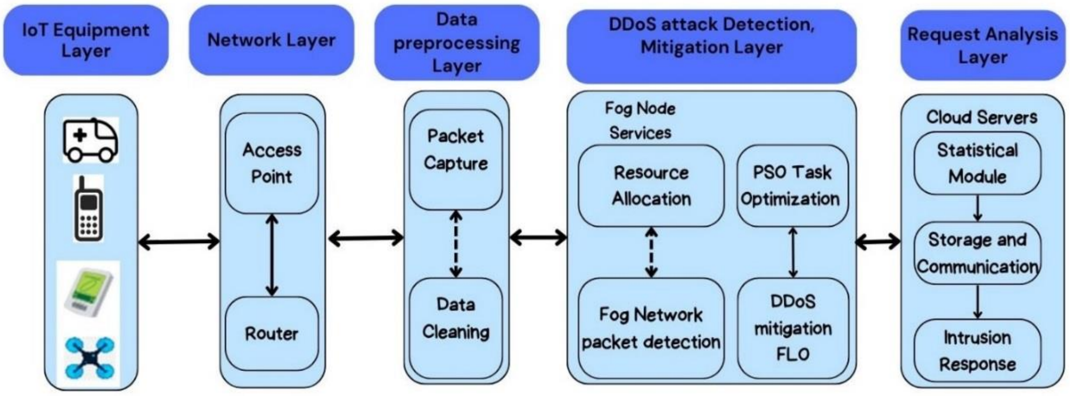


Figure 1: The allocation of computing resources in the IoT-Fog cloud server architecture [7]

The main steps of our proposed system are explained as follows:

♦ IoT devices initiate requests to access particular types of applications, such as HTTP requests, UDP requests, and FTP file upload requests on cloud servers.

♦ A request has been transmitted to fog nodes for the purpose of redirecting it to a designated cloud server.

♦ Request management is used on the fog layer (basebroker and fog node) to filter and recognize request types using packet capture and data cleaning pre-processing methods. It also defines what a normal request looks like by matching log file configuration records with incoming data and teaching the model how to tell the difference between normal and abnormal requests.

♦ Fog nodes periodically transmit Beacon BFRA request packets to cloud servers, resulting in improved value with respect to processing costs and server specifications for each active server. Additionally, fog nodes continue to transmit check packets at regular intervals to remain updated on the latest server state and to assess the optimal server for processing subsequent requests.

♦ Building network statistics for all system cases and an analysis file as created as a CSV dataset (DDoS-IoT), which contains all network evaluation results registered through the running time.

♦ The created DDoS-IoT dataset contains network parameters for both normal and abnormal behavior. It was passed into the Java toolbox to classify and mitigate DDoS attacks. After analysis, the results were trained with three algorithms: MLP, KNN, and SVM.

However, Figure 2 shows the proposed DDoS attack mitigation approach.

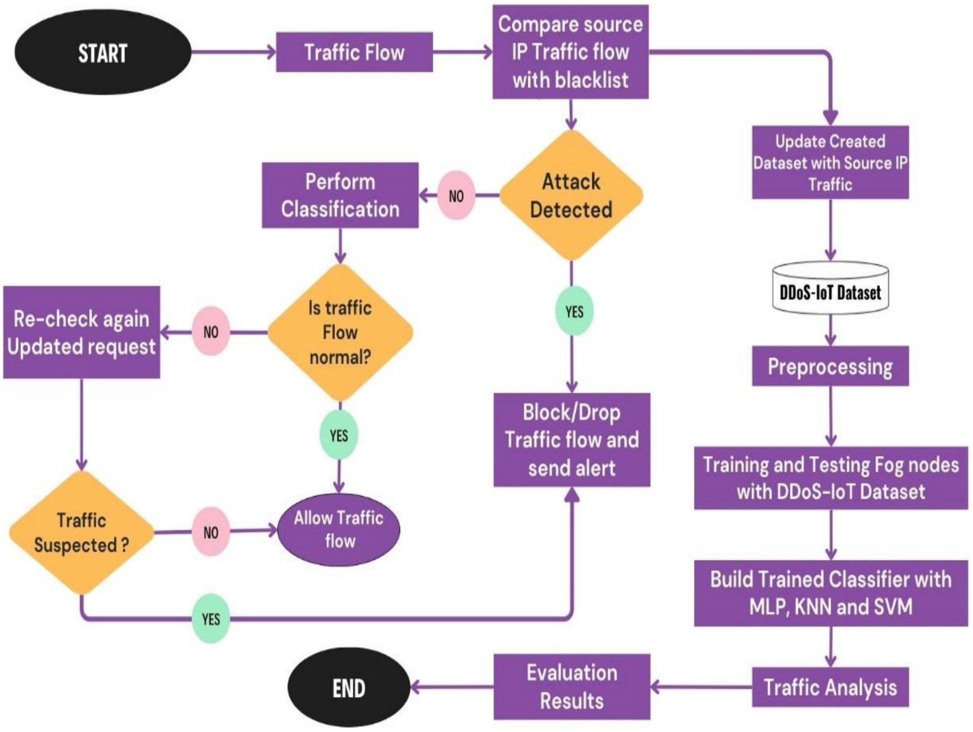

START
Traffic Flow
Compare source IP Traffic flow with blacklist
Update Created Dataset with Source IP Traffic
Perform Classification
NO
Attack Detected
YES
DDoS-IoT Dataset
Re-check again Updated request
NO
Is traffic Flow normal?
YES
Preprocessing
Block/Drop Traffic flow and send alert
Training and Testing Fog nodes with DDoS-IoT Dataset
Traffic Suspected ?
NO
Allow Traffic flow
YES
Build Trained Classifier with MLP, KNN and SVM
END
Evaluation Results
Traffic Analysis

Figure 2: The proposed method for DDoS attack mitigation using classification algorithms

The suggested system is based on the following key components:

3.1 DDoS Attack Mitigation Model

The suggested model has established an anomaly mitigation system for the IoT network. This has been achieved through the design and implementation of a DDoS attack detection system. The system relies on a statistical approach that incorporates three algorithms: MLP, KNN, and SVM. The paper's contributions might be regarded as significant within the realm of IoT devices, as the implementation of anomaly mitigation algorithms poses challenges due to resource constraints, so the proposed system employed a PSO algorithm for resource allocation in the IoT network. The objective is to construct a value-based fitness leader optimization (FLO) by achieving the performance of the system, which exhibits a commendable level of efficacy in terms of achieving a high rate of detection and accurately classifying network flows as either normal or abnormal, generated with and without resource allocation. It is implemented in a centralized and distributed Fog-Server server environment; each Fog node periodically evaluates the network resources to determine the optimal Fog-Server for processing incoming requests. The primary objectives of the proposed system under consideration are:

- Decrease malicious activity: It is applied by detecting an unauthorized intrusion into a computer system or network perpetrated by someone with malicious intent and filtering the source IP of the malicious node.
- DDoS attack mitigation: It is based on the Fitness Leader Optimization (FLO) method to control network traffic by fog nodes.
- Decrease Cost: The cost incurred to process the task in the IoT constrained environment.
- High Scalability: This shows the capability of the resource allocation mechanism, which can be applied to the machines and the tasks.
- High Flexibility: The joining of new IoT nodes and the revocation of the node mechanism are flexible.
- Maximum Resource Utilization: IoT resource utilization is maximized with the Fog-Server system.
- Decreased Processing Time: The total time it takes to execute a service request from IoT is minimized.
- Increase availability: The capability of the computing system can be taken up to maintain the system’s performance.
- High throughput: It is done through the service requests that are processed in the Fog-Server system.

The Fog node is responsible for monitoring network-level activity and analyzing network traffic flow in order to identify potential attacks or aberrant behaviors.

The efficacy of the DDoS mitigation system that is based on anomaly detection is contingent upon the establishment of a comprehensive profile that accurately represents the typical patterns of traffic flow. The comparison is made between the flow of network traffic and the established normal profiles, whereby any deviation from these patterns is classified as a DDoS assault. The anomaly-based intrusion detection system (IDS) possesses the capability to identify and detect zero-day or previously unidentified assaults. The development of this method is commonly achieved through the utilization of statistical or machine learning approaches. Despite being well-suited for implementation in the IoT environment, this technology has its own set of obstacles, notably a significant occurrence of false-positives when network traffic flow is misclassified. In order to address the issue of resource scarcity on IoT devices. The utilization of the fog computing concept is possible. The purpose of edge computing is to enhance the quality of services for time-sensitive applications by deploying resources such as storage, compute, and network services closer to the network edge. The proposed approach is designed to effectively implement anomaly mitigation strategies in IoT networks, therefore relieving IoT devices of the computational and storage demands they now face. This will guarantee the effective functioning of the anomaly mitigation strategies.

As previously stated, the used system also possesses the capability to demonstrate the significance of fog computing within anomaly detection frameworks. This is achieved by its ability to locally monitor network traffic, thereby facilitating the identification of diverse threats at the most fundamental level. The pseudocode of the Fitness Leader Optimization (FLO) approach is demonstrated in Algorithm 1.

Algorithm 1: Proposed Fitness Leader Optimization (FLO) algorithm of anomaly DDoS mitigation

```
Input: Traffic flow
Output: Attack detected
Step1: Configure Traffic Flow (Tf) with source / destination IPs
Configure BlackList (BL)
Step2: Compare source IP (TF) with (BL)
Get Input traffic
IF source IPs within (BL) THEN
    Block / drop (TF)
    Send alert
Else
    Proceed to Step3;
End
Step3: perform Classification
Is Normal or Suspicious or Attack
IF TF = Attack THEN
    Block / drop (TF)
    Send alert
Else
    Go to Step3;
    End
Start classification model (MLP, KNN, SVM) development
Input: TF
Output: Normal or Suspicious or Attack
IF T= Attack THEN
    Block / drop (TF)
    Send alert
    Update database
Else
    IF TF= Normal Then
        Allow Traffic flow
End
IF TF= Suspicious THEN
    Allow Traffic to pass to Second Check Point
Else
End
IF T= Suspicious THEN
    Block/drop Traffic Flow
    Update database
Else
Allow Traffic flow
End
```

End of algorithm

3.2 Resource Allocation Model with PSO

The utilization of PSO as a meta-heuristic approach for resource allocation in IoT fog computing networks has been proposed. This approach aims to improve the efficiency, search speed, and cost effectiveness of IoT resource-constrained devices.

The main steps of using the PSO meta-heuristic approach implementation are presented as follows:

⊕ Evaluating current fitness.

⊕ Determining whether the present position represents optimal individual performance.

⊕ Updating new particle velocity.

⊕ The present paper involves an assessment of the constant inertia weight for current weight evaluation, which pertains to the determination of the appropriate weight to assign to the previous velocity.

⊕ Revise the position of the particle in accordance with the updated velocity modifications.

⊕ Adjust maximum fog position if necessary.

⊕ Determining the best fog position for the request.

⊕ Establishing the swarm.

⊕ Beginning the optimization loop.

⊕ Cycle through particles in a swarm and evaluate their fitness.

⊕ Determining if the current fog particle is the best.

⊕ Cycle through the swarm and update velocities and positions.

The PSO algorithm aims to allocate incoming tasks in a distributed system to available computing resources based on predetermined criteria and objectives, such as the number of incoming requests and the required makespan time to complete the tasks. The PSO algorithm being examined employs the portrayal of every individual in the swarm as a feasible solution to the optimization problem at hand. Throughout the optimization process, each particle experiences a procedure of updating that considers both the global best particle’s position and its own local best position.

The allocation of particle resources is assessed by Basebroker through an analysis of the behavior and requirements of both the fog nodes and cloud servers. The Fog platform facilitates the identification and selection of suitable resources. The basebroker allocates resources in accordance with the behavior of IoT devices. The system facilitates the allocation, tracking, booking, and administration of resources within the IoT-Fog cloud server computing framework. Furthermore, it oversees the resources that are accessible within the Fog computing environment.

3.3 Classification Model

A machine learning model has the capability to approximate the quantitative behavior of a system [23]. The quantitative findings derived from statistical analysis may be effectively compared to experimental data in order to assess the strengths and limits of the model. Accuracy and the false-positive rate are crucial factors in evaluating the effectiveness of assaults [24].

If many classifiers are used to test a model, this model will be more comprehensive and accurate, and it can be generalized in the future [25]. For all the above reasons, different classifiers have been applied, ranging from MLP to KNN to SVM. In fact, each one is implemented individually.

3.3.1 MLP

Multilayer Perceptron (MLP) is a subcategory of artificial neural networks (ANN) that comprises a feed-forward neuron to form a feed-forward neural network. It has three layers' types: input, hidden, and output layers. MLP takes an input from a previous layer, then executes the non-linear transformation through the hidden layers [26]. Back propagation (BP) is a substantial algorithm for MLP training. It is able to analyze the errors and optimize each value of weight depending on the errors [27].

To compute the weight, assume the neural network has (m) neurons; this neuron is driven by the input vector X_n, and n indicates the time step of the iterative process, which involves the adjusting step of the input weights $_{(mi)}$. Accordingly, every data sample passes through the training step containing $X_{(n)}$ and its output referred by $d_{(n)}$ [28]. Thereafter, it generates an output that is denoted by $_{(n)}$ and calculated by Eq. (1):

$$y_{m(n)} = f \sum_{i=1}^{j} x * w_{(mi)} \tag{1}$$

Where f indicates an activation function. This output is compared with the target output $_{(n)}$ and the error $_{(n)}$ can be computed according to Eq. (2):

$$e_{m(n)} = (\, d_{m(n)} - y_{m(n)}) \tag{2}$$

3.3.2 KNN

K-Nearest Neighbor (kNN) is an essential statistical method and one of the simplest forms that is applied for classification purposes. This algorithm uses the Euclidean distance as a metric to specify the distance between two points, samples, nodes, etc. [29]. The KNN algorithm is able to predict new data sample values and then classify new instances based on the (k) value of their nearest neighbors. This value is normally a positive integer [30].

The major steps of the KNN algorithm are described as follows [31]:

A. Load the dataset samples.

B. Specify the (K) value.

C. For each data sample:

- Find the Euclidean distance.
- Sort all computed distances.
- Choose the first k data.
- Assign a class to the data sample based on the majority of classes present in the chosen data samples.

D. End.

3.3.3 SVM

SVM is a well-known method that is used mainly in classification tasks. However, this algorithm applied to the nonlinear function estimation and its problems by projecting the dataset samples into higher-dimensional space [32].

The objective of SVM is to find the best possible line, or decision boundary, that detaches the dataset samples into different classes [33]. In the case of complex data, SVM is beneficial to analyze this data type because this method is able to transform the data into a higher-dimensional space, making it easier to find the boundaries [34].

The proposed algorithms classify the created dataset into normal traffic and abnormal DDoS traffic. The DDoS-IoT dataset was made from the network simulation statistics that were recorded for each IoT device parameter for 600 minutes. The exported CSV file has all the simulation attributes for dividing the system results into two main groups: normal data traffic and abnormal traffic, along with two main subgroups, as shown in Figure 3.

The dataset effectively represents the characteristics of traffic flow in the specific scenario where the framework is intended to be implemented. The dataset will be utilized for the experimental procedures of trained machine learning classifiers such as MLP, KNN, and SVM.

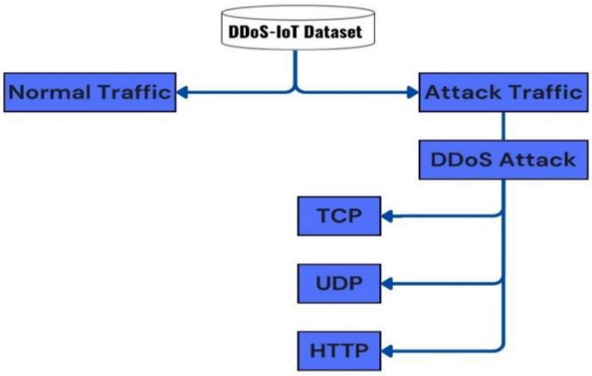


Figure 3: Classification of the DDoS-IoT dataset [23]

The trained classifier is tested 10 times to discover abnormal behavior for DDoS attack nodes and mitigate threats to the network elements and services it provides. In the pre-processing stage of data classification, the created dataset is converted into various formats for efficient classification and detection of DDoS to assess the performance of the machine learning classifiers inside the suggested framework. A sample size of 3,112 occurrences was randomly selected from the DDoS-IoT dataset. Fifteen features were chosen based on the utilization of the ranker search approach and knowledge gained in the feature selection process, effectively eliminating duplicate and unnecessary characteristics. The selected features exemplify the optimal characteristics that are expected to yield high performance in classification tasks. Table 1 shows the used features of the DDoS-IoT dataset attributes.

Table 1: The used DDoS-IoT dataset attributes

Seq.	Features	Description
1.	Src_ddr	IP source
2.	Src_port	Source port number
3.	Dst_ddr	Destination IP
4.	Dst_port	Dest Port number.
5.	Rcrd_dur	Total duration
6.	Snd_dev	average record variance
7.	N_IC_p_IP	Source IP-inbound connections
8.	Mn_dur_R	minimal record retention
9.	F_st_No	numerical state representation
10.	A_Mn_R	average record length
11.	N_IC_p_Dst_IP	Destination IP inbound connections
12.	Dst_spS	destination-to-source packets/second
13.	Src_Dsts	Source-to-destination packets/second
14.	Mx_R	maximum aggregated record length
15.	Attack_Class	class label: DDoS 1 and regular traffic 0.

4. Results

The suggested IoT-fog-cloud server scenario for mobility uses three fog nodes to manage traffic and reduce the huge number of requests sent by malicious nodes (DDoS attack nodes) in the network. Each of the fog nodes sends the incoming request from an optimized base broker to the selective mobility cloud server, taking into account how busy each cloud server is. In addition, IoT connected with network elements as an access point, gateway router, base broker, three fog nodes, and three cloud servers. Table 2 displays the technical details of the network components that impact the IoT resource allocation. These details include the network setting.

Table 2: The proposed Fog-Cloud Server System Specifications

Node IP	Behavior	Network Interface	Local port
20.20.20.2/30	IoT-Node1	Wireless connection with Mobility services	1120
20.20.20.6/30	IoT-Node2		1125
20.20.20.22/30	IoT-Node3		1126
20.20.20.30/30	IoT-Node4		1127
20.20.20.14/30	IoT-Node5		1128
20.20.20.18/30	IoT-Node6		1129
20.20.20.19/30	IoT-Node7		
20.20.20.21/30	IoT-Node8		
20.20.20.46/30	Fog Node1	Eth1_to_etch3/ Mobility services	1130
20.20.20.30/30	Fog Node2		1133
20.20.20.62/30	Fog Node3		1136
20.20.20.45/30	Server1		1132
20.20.20.29/30	Server2		1138
20.20.20.53/30	Server3		1140
/	Access Point	/	/
20.20.20.17/30	Gateway Router	Eth0_to_eth1	1134
20.20.20.1/30	BaseBroker (virtual)	Eth0_to_eth3	1135

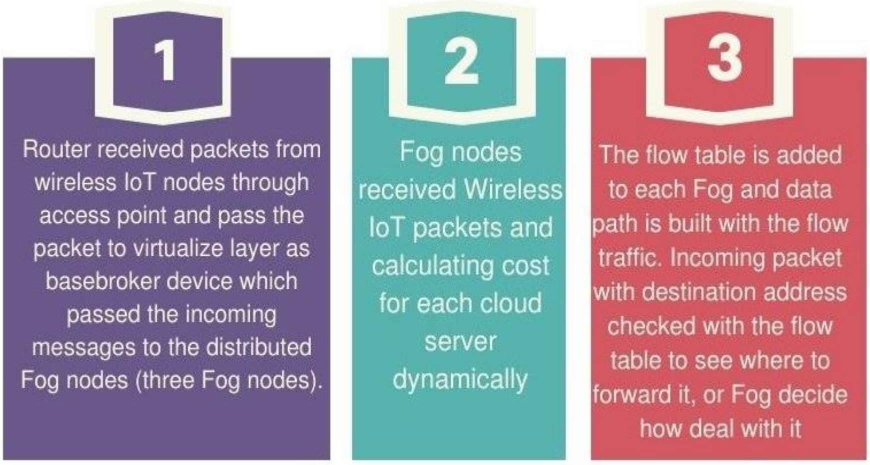
Table 3 shows the network simulation parameters to implement DDoS attack mitigation and classification of traffic into the IoT network as normal and abnormal traffic.

Table 3: Network simulation parameters

Seq.	Parameter	Value
1.	Number of cloud server nodes	3
2.	Number of fog nodes	3
3.	Number of IoT devices	8
4.	Average behavior detection time MLP for 10 samples	9516 ms
5.	Average behavior detection time KNN for 10 samples	3 ms
6.	Average behavior detection time SVM for 10 samples	577 ms
7.	Number of simulation runs	10
8.	Number of data instances	3112
9.	Number of data attributes	15

Firstly, the network setting is established by the implementation of a distributed Fog-cloud server allocation of resources management system that leverages network servers to effectively manage network congestion. The optimization process happened between the basebroker and fog node using the meta-heuristic PSO algorithm, with the main setting being:

Step 1: **general setting** for initializing position, number of iterations, current population of particles, and velocity
Step 2: **network configuration,** which is used for network configuration with local port number, interfaces, message length, packet type, and time management with synchronized task management.
Step 3: **fog server** for building system topology with the mobility feature of fog server elements and with 8 IoT nodes.
Step 4: **resource allocation** for distributed Fog-Cloud server configuration with the network setting for wireless IoT nodes.
Step 5: **update step** for exchanging setting messages from wireless IoT with access point and gateway router to build network table and update IoT locations with the main phases. Indeed, the main phases of the updating step (optimization step) are depicted in Figure 4.



1 Router received packets from wireless IoT nodes through access point and pass the packet to virtualize layer as basebroker device which passed the incoming messages to the distributed Fog nodes (three Fog nodes).

2 Fog nodes received Wireless IoT packets and calculating cost for each cloud server dynamically

3 The flow table is added to each Fog and data path is built with the flow traffic. Incoming packet with destination address checked with the flow table to see where to forward it, or Fog decide how deal with it

Figure 4: Optimization update step in wireless IoT network configuration

Additionally, the distributed fog computing system that has been suggested is executed by employing a resource allocation optimization algorithm. This algorithm aims to achieve load balancing, reduce response time, and enhance resource utilization. The primary steps involved in this process are as follows:

A. The IoT node initiated requests to the servers to process their request as HTTP web access or file uploading service through network gateway and basebroker, then Fog nodes as the network elements such as an access point, gateway router, and basebroker configured to pass data messages to the Fog nodes layer.
B. Fog nodes receive requests from IoT, modify flow rules, balance the load among all available connected servers, and estimate the current cost of each cloud server.
C. Fog nodes periodically transmit ICMP echo messages to the present servers in order to ascertain the most recently updated cost value (load) of the servers. The server that responds with the lowest processing cost is deemed to be the optimal server.
D. The fog nodes assess the task statuses of five individual servers and contemporaneously revise the cost fitness leader optimization (FLO) method based on the characteristics of connection traffic and the degree of overloading present in each of the operational servers.

E. Fog nodes depend on the incoming request cost to distribute the request, in some cases of high-overloaded data, to multiple servers to reply to the response to each request by the IoT nodes.

4.1 The results of Distributed Fog Computing (Case 2)

The suggested approach is based on four discrete case studies. The first case study does not involve the allocation of resources for fog server network services. The second case study utilizes the PSO algorithm for resource allocation. The third case study applied a DDoS attack that was simulated on the network. Lastly, the fourth case study utilizes a DDoS attack mitigation approach. This approach is implemented in each fog node used in distributed fog computing in order to optimize and schedule incoming requests to specific fog nodes and mitigate DDoS attack effectiveness on IoT devices. The resource allocation algorithm used to manage tasks in processing (active requests) that are executed on the Fog also has different directions to specific cloud servers. Therefore, in order to acquire an optimal choice of fog, it is required to discover an ideal way to place the various responsibilities evenly on the various fogs, taking into consideration the various properties of the fog and the application.

4.2 Fog Computing System without Resource Allocation

The strategy for implementing overload is applied to simulate the state of higher computational processes without any allocation of resources, and traffic is high with makespan and a high required time as a fitness value, which requires allocating resources to manage data requests among available cloud servers.

Table 4 presents information regarding the allocation of resources, processing time, and execution cost for IoT nodes. The table also highlights the benefits that IoT devices receive from channel utilization. Additionally, the table offers information on how long packets remain in the queue before the servers process them. The fitness value is maximized as the total sum is 0.3442 seconds without resource allocation in a distributed fog server environment.

Table 4: Evaluation metrics without resource utilization in a distributed fog-server system

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value in Sec
IoT Node	IoT-Node1	0.145267	26.4537	27.29895	0.019741
	IoT-Node2	9.626573	2.5194	2.599933	0.077691
	IoT-Node3	1.093248	1.2597	1.654713	0.017889
	IoT-Node4	10.62847	34.0119	34.23962	0.014545
	IoT-Node5	0.012131	35.2716	35.56856	0.02052
	IoT-Node6	10.77373	15.1164	15.18493	0.061275
	IoT-Node7	10.62847	34.0119	34.24793	0.014089
	IoT-Node8	0.012131	25.5816	25.88905	0.019855
	Server1	10.17919	25.194	/	0.080665
	Server2	0.06245	68.0238	/	0.007515
	Server3	11.28317	3.270375	/	0.01045
Fog (average all interfaces)		5.74651	25.62943	24.34796	/

The uploaded and downloaded files are displayed in Table 5 for the servers with different file sizes to measure the speed of data transfer and makespan as the required time as the end time of the last job to complete the entire process.

Table 5: FTP file upload/download from cloud server 1 to server 3 without allocation in the distributed fog-server system

FTP File Size Upload/Download	Speed in Mbps	Makespan in Sec
	Server 1: 20.20.20.45/30	
512 KB	1.12837	1.728638
5 MB	2.3096	5.508996
16 MB	2.3184	86.20251
	Server 2: 20.20.20.29/30	
512 KB	1.3575	1.755968
5 MB	2.4065	5.525589
16 MB	2.3889	86.40846
	Server 3: 20.20.20.53/30	
512 KB	1.2605	1.800868
5 MB	2.3625	5.513876
16 MB	2.3801	22.08479

4.3 Resource allocation with Meta-heuristic PSO algorithm

The second case study presents a proposed system that utilizes a fog cloud server architecture. This system employs the Particle Swarm Optimization (PSO) algorithm for resource allocation to effectively manage workload distribution and balance the overload of resources within the IoT-Fog-Cloud server computation environment by distributing incoming IoT request packets to the optimal cloud server while minimizing the total makespan, fitness value and execution cost.

Table 6 shows how channel resources are distributed, how long it takes to process, how much it costs to run, and how long it takes for IoTs to reach their fitness value. The results of how resources are distributed in fog computing were better than those of the first case study. This is due to the increase in channel availability and resource utilization, as well as the decrease in processing time and execution cost. Total fitness is 0.264636, so it is better with decreased fitness value compared with the state without resource allocation, and it is also better compared with the PSO of a centralized system due to the increase in available cloud servers after task scheduling in a distributed system.

Table 6: PSO case study channel allocation in a distributed fog-server system

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value in Sec
IoT Node	IoT-Node1	0.164263	19.84028	17.35707	0.014966
	IoT-Node2	10.88543	1.88955	1.938902	0.059035
	IoT-Node3	1.236211	0.944775	1.186754	0.013588
	IoT-Node4	12.01835	25.50893	20.64982	0.011159
	IoT-Node5	0.013716	26.4537	21.63507	0.015541
	IoT-Node6	12.1826	11.3373	9.37958	0.04687
	IoT-Node7	12.01835	25.50893	23.65276	0.010647
	IoT-Node8	0.013716	19.1862	15.37022	0.014737
	Server1	11.51031	18.8955	/	0.064513
	Server2	0.070616	51.01785	/	0.005558
	Server3	12.75866	2.452781	/	0.008022
Fog (average all interfaces)		6.497977	19.22207	/	/

Table 7 presents both upload and download files to the available servers with a lower fitness value and a different file size, and it shows that the PSO is useful for file uploading requests through a smaller makespan and acceptable speed compared with the 1st case study without resource allocation and with the PSO of a centralized fog-server system.

Table 7: FTP file upload/download from server 1 to server 3 of the PSO of the distributed fog-server system

FTP File Size Upload/Download	Speed in Mbps	Makespan in Sec
	Server 1: 20.20.20.45/30	
512 KB	1.3540	1.358959
5 MB	2.7715	4.330867
16 MB	2.7821	67.76764
	Server 2: 20.20.20.29/30	
512 KB	1.6306	1.380445
5 MB	2.8906	4.343912
16 MB	2.8694	67.92954
	Server 3: 20.20.20.53/30	
512 KB	1.5141	1.415742
5 MB	2.8376	4.334704
16 MB	2.8588	17.36184

4.4 Results of DDoS Attack

The DdoS attack case study is simulated in the proposed system to generate a huge amount of data packets from malicious nodes in a short period of time, flood the network with unwanted packets, and increase the waiting time of IoT nodes, which led to exhausted IoT resources and disconnected them. The data traffic is stored and analyzed in the CSV DdoS-IoT dataset. The next step in classification with machine learning algorithms is MLP, KNN, and SVM. Total fitness value is maximized at 0.671259 seconds for IoT devices due to the increased total required time to complete any task in a DdoS attack environment. Table 8 displays the main simulation results of the DdoS attack case study.

Table 8: Evaluation metrics of DdoS attacks in distributed fog-server systems

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value in Sec
IoT Node	IoT-Node1	0.08716	48.93935	51.86801	0.038495
	IoT-Node2	IoT-Node1	4.66089	5.239873	0.151497
	IoT-Node3	IoT-Node1	2.330445	4.343955	0.034884
	IoT-Node4	IoT-Node1	62.92202	65.05528	0.028363
	IoT-Node5	IoT-Node1	65.25246	67.58026	0.040014
	IoT-Node6	IoT-Node1	27.96534	28.85137	0.119486
	IoT-Node7	IoT-Node1	62.92202	65.07107	0.027474
	IoT-Node8	IoT-Node1	47.32596	49.1892	0.038717
	Server1	6.107514	46.6089	/	0.157297
	Server2	0.03747	125.844	/	0.014654
	Server3	6.769902	6.050194	/	0.020378
Fog (average all interfaces)		3.447906	47.41445	46.26112	/

The files that were uploaded and downloaded are displayed in Table 9 on the cloud servers with different file sizes from normal and malicious nodes to measure the speed of data transfer, and makespan is the required time for the last job to complete the entire process.

Table 9: FTP file upload/download from server 1 to server 3 of the DDoS attack in a distributed fog-server system

FTP File Size Upload/Download	Speed in Mbps	Makespan in Sec
	Server 1: 20.20.20.45/30	
512 KB	0.6431	3.284412
5 MB	1.3164	10.46709
16 MB	1.3215	163.7848
	Server 2: 20.20.20.29/30	
512 KB	0.7738	3.31878
5 MB	1.3717	10.44336
16 MB	1.3617	163.312
	Server 3: 20.20.20.53/30	
512 KB	0.7185	3.385632
5 MB	1.3466	10.36609
16 MB	1.3566	41.51941

4.5 Results of DDoS Mitigation System

The proposed mitigation system is implemented within a fog node to effectively manage network traffic. This is achieved by employing a model trained and tested using Particle Swarm Optimization (PSO) for resource allocation and machine learning techniques for classifying traffic into normal and abnormal categories. As a result, the system is able to disconnect source IP packets that generate high traffic flow. The approach used is based on the total size of generated packets, the number of packets, and the required time. It trained 10 times to identify a normal case study from the DDoS case study, and all results exported into a dataset that was evaluated with machine learning were MLP, KNN, and SVM. In this case study, the total fitness value of 0.284556 is minimized compared with the DDoS attack case study, and other evaluation metrics are enhanced compared with the DDoS attack case study in Table 10.

Table 10: DDoS attack mitigation case study evaluation results

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value in Sec
IoT Node	IoT-Node1	0.151122	18.25306	18.93208	0.016093
	IoT-Node2	10.0146	1.738386	1.803179	0.063479
	IoT-Node3	1.137314	0.869193	1.103681	0.014611
	IoT-Node4	11.05688	23.46822	23.85433	0.011999
	IoT-Node5	0.012619	24.3374	24.77062	0.016711
	IoT-Node6	11.20799	10.43032	10.58301	0.050398
	IoT-Node7	11.05688	23.46822	23.85707	0.011448
	IoT-Node8	0.012619	17.6513	18.0143	0.015846
	Server1	10.58949	17.38386	/	0.069369
	Server2	0.064967	46.93642	/	0.005976
	Server3	11.73797	2.256559	/	0.008626
Fog (average all interfaces)		5.978139	17.6843	/	/

Table 11 demonstrates upload and download files to the available servers with a lower fitness value for different file sizes, and it shows that the PSO is useful for file uploading requests.

Table 11: FTP file upload/download from server 1 to server 3 of the DDoS attack mitigation system

FTP File Size Upload/Download	Speed in Mbps	Makespan in Sec
	Server 1: 20.20.20.45/30	
512 KB	1.2728	1.508444
5 MB	2.6052	4.807262
16 MB	2.6152	75.22208
	Server 2: 20.20.20.29/30	
512 KB	1.5327	1.532294
5 MB	2.7171	4.821742
16 MB	2.6972	75.40179
	Server 3: 20.20.20.53/30	
512 KB	1.4232	1.656418
5 MB	2.6674	5.071604
16 MB	2.6873	20.31335

As depicted in Figure 5, the transfer of total utilization of resources and fitness value during the execution of the transmission position for the utilized case studies was demonstrated.

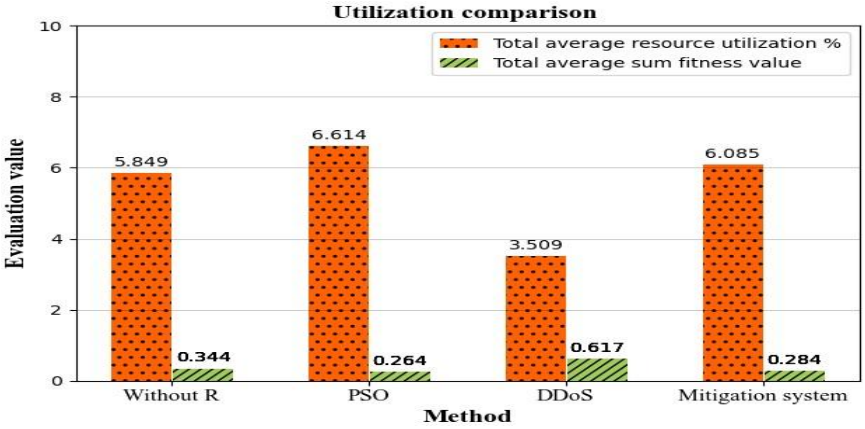


Figure 5: The proposed distributed fog system comparison with the average of resource utilization and fitness time

The findings indicate that the suggested PSO strategy outperformed the other cases of resource allocation. The PSO method exhibits an average total utilization of resources of 6.614% and a superior total sum fitness value of 0.264. On the other hand, Figure 6 illustrates the required time and the execution cost of the proposed system, as well as others.

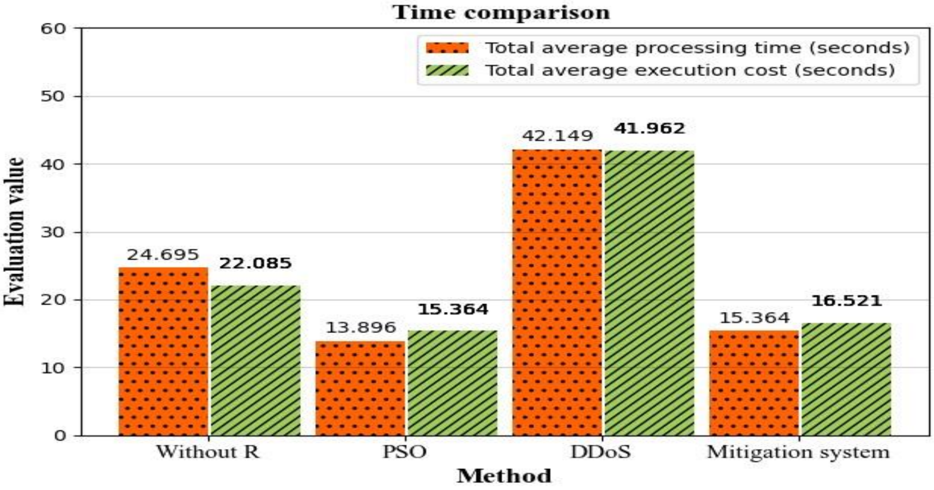


Figure 6: The proposed distributed fog system comparison with average processing time and execution cost.

4.6 Results of Machine Learning classification model

The proposed DDoS attack classification model is based on three algorithms: MLP, KNN, and SVM, to classify data traffic into normal and abnormal in the proposed IoT fog computing network. The main accuracy details of the proposed case study using data mining and machine learning on the DDoS-IoT dataset are: SVM has a high accuracy of 99.6784% and a time to build model of 577 ms, while MLP has a level two accuracy of 99.6784% DT and a time taken to build model of 9516 ms. Besides, KNN has a low result accuracy of 99.5177%, and the time taken to build the model is 3 ms. Table 12 shows the accuracy and time details of the evaluated parameters of the DDoS-IoT dataset.

Table 12: The results of machine learning for DDoS-IoT data analysis

Method Name	Accuracy	Time
Support Vector Machine	99.6785 %	577 ms
MLP Neural	99.6784 %	9516 ms
K-Nearest Neighbor(KNN)	99.5177 %	3 ms

Furthermore, the evaluation criteria used in the proposed system, such as mean absolute error (MAE), root mean squared error (RMSE), and error rate shown in Table 13, show the DDoS attack detection of the evaluation criteria of the proposed algorithms. The SVM is 0.00321, which is almost lower than the MAE value, so it is the best compared with others. The SVM results of the RRSE statistic are (97.354%), which indicates that the lower is the best compared with other algorithms. In addition, the SVM error rate is the best for the DDoS-IoT dataset. The AUC parameter shows that the MLP is the best model for predicting attack classes (1 DDoS attack, 0 normal).

Table 13: MAE and RRSE for the DDoS-IoT machine learning

Evaluation Criteria	Predication		
	SVM	MLP	KNN
Mean Absolute Error(MAE)	0.00321	0.01992	0.00518
Root Relative Squared Error(RRSE)	97.3754	106.2161	119.2218
Error Rate	0.0032	0.00323	0.00482
Area Under the Curve (AUC)	0.00321	0.00749	0.00258

There are also other evaluation classifiers based on the three machine learning classifiers, as shown in Table 14 of the proposed system. These are used for both the normal case without DDOS-IoT (class 0) and the case with DDOS-IoT (class 1). SVM precision of 99.67845% can be seen as a measure of high quality that returns more relevant results than irrelevant ones.

Table 14: Evaluation Details of the Presence of DDOS-IOT in Machine Learning for Big Data Analys**is**

Evaluation Parameters	Machine Learning Algorithms		
	SVM	MLP	KNN
Precision	99.67845	99.6784	99.5176
Detection Rate (DR)	1	1	0.99838
False Alert Rate (FAR)	1	1	1

Upon comparison with other relevant works, the proposed system exhibited superior evaluation outcomes, as evidenced by the data presented in Table 15.

Table 15: The proposed distributed fog-server system comparison with the other related works

Ref.Year	Algorithm	Simulation Tool	Total Makespan in Sec
[35], 2021	Extended Particle Swarm Optimization (EPSO)	iFogSim	342.53
[36], 2022	Particle Swarm Optimization (PSO)	iFogSim	220
The proposed PSO		FogNetSim++, iFogSim extension	170.223

The results compare with the other related works, as shown in Table 16. The best accuracy result of the proposed system of all case studies is shown in the cases of Naïve Bayes, Decision Tree, and SVM at 98.646%.

Table 16: The results of the DDoS-IoT dataset analysis with the other systems

Ref.Year	Dataset, Architecture	AI Technique	Accuracy
[17], 2023	Bot-IoT dataset, DDoS-IoT-Fog Architecture	Exponentially Weighted Moving Average (EWMA)	88.0 %
		K-nearest neighbors (KNN)	98.0 %
		Cumulative Sum Algorithm (CUSUM)	88.0 %
		Integration (EWMA, KNN, CUSUM)	99.0 %
[19], 2023	ToN-IoT dataset, DDoS-IoT Architecture	Non-Dominated Sorting Genetic Algorithm (NSGA-II)-based meta-heuristic with SVM	98.86 %
		Hybrid Filter + NSGA-II with SVM	99.48 %
The proposed system	DDoS-IoT dataset, DDoS-IoT-Fog Architecture	SVM	99.67 %
		MLP	99.67 %
		KNN	99.51 %

5. Conclusion

One of the biggest threats faced by IoT networks in contemporary times is the DDoS assault. Numerous systems for detecting IoT DDoS attacks mostly depend on internally generated datasets, as the availability of publicly accessible datasets pertaining to such attacks is limited owing to apprehensions over privacy and security. When a detection and mitigation system is used with a dataset that was made with an incorrect representation of the packet or flow and includes unqualified characteristics, a lot of false alarms are raised. The objective of this paper is to develop a methodology for allocating resources in an IoT network and generate a dataset of DDoS assaults. This dataset can then be utilized to fine-tune, benchmark, and evaluate the effectiveness of any detection and mitigation systems specifically built for identifying DDoS attacks. The datasets under consideration satisfy the requirements necessary for a high-quality dataset, hence guaranteeing their utility to other authors. In addition, the distributed fog-server computing framework under consideration has been implemented through four distinct case studies. The results indicate that the suggested system has enhanced the network’s performance by ensuring the effective handling of IoT requests through the utilization of distributed services as an optimizer for PSO fog-server computing. In addition, three classifiers were selected, exhibiting differences in terms of their kind, classification performance, and the quantity of characteristics employed. The outcomes obtained were as follows: The SVM achieved a classification accuracy of 99.6785%, followed closely by the MLP with a performance of 99.6784%. The KNN algorithm achieved a slightly lower accuracy of 99.5177%. Experimental tests have shown that the suggested dataset can effectively capture attack traffic, showing a high level of detection accuracy while maintaining a low rate of false positives.

References

[1] S. Rani, H. Babbar, G. Srivastava, T. R. Gadekallu and G. Dhiman, "Security Framework for Internet-of-Things-based Software-Defined Networks using Blockchain," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 6074-6081, Apr. 2022. Doi: 10.1109/JIOT.2022.3223576.

[2] K. F. Hassan and M. E. Manaa, "Detection and mitigation of DDoS attacks in internet of things using a fog computing hybrid approach," *Bull. Electr. Eng. Informatics*, vol. 11, no. 3, pp. 1604-1613, Jun. 2022. Doi: 10.11591/eei.v11i3.3643.

[3] F. Al-Turjman, H. Zahmatkesh and R. Shahroze, "An overview of security and privacy in smart cities' IoT communications," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, pp. 1-20, Jun. 2019. Doi: 10.1002/ett.3677.

[4] A. I. Awad, M. M. Fouda, M. M. Khashaba, E. R. Mohamed and K. M. Hosny, "Utilization of mobile edge computing on the Internet of Medical Things: A survey," *ICT Express*, vol. 9, no. 3, pp. 473-485, Jun. 2023. Doi: 10.1016/j.icte.2022.05.006.

[5] A. N. Kadhim and M. E. Manaa, "Design an efficient internet of things data compression for healthcare applications," *Bull. Electr. Eng. Informatics*, vol. 11, no. 3, pp. 1678-1686, Jun. 2022. Doi: 10.11591/eei.v11i3.3758.

[6] A. M. Aslam, R. Chaudhary, A. Bhardwaj, I. Budhiraja, N. Kumar and S. Zeadally, "Metaverse for 6G and Beyond: The Next Revolution and Deployment Challenges," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 32-39, Mar. 2023. Doi: 10.1109/IOTM.001.2200248.

[7] A. Ometov, O. L. Molua, M. Komarov and J. Nurmi , "A Survey of Security in Cloud, Edge, and Fog Computing," *Sensors*, vol. 22, no. 3, pp. 1-27, Jan. 2022. Doi: 10.3390/s22030927.

[8] S. Padhy, M. Alowaidi, S. Dash, M. Alshehri, P. P. Malla, S. Routray and H. Alhumyani, "AgriSecure: A Fog Computing-Based Security Framework for Agriculture 4.0 via Blockchain," *Processes*, vol. 11, no. 3, pp. 1-27, Mar. 2023. Doi: 10.3390/pr11030757.

[9] J. B. JR, B. Costa, L. R. Carvalho, M. J. F. Rosa and A. Araujo, "Computational Resource Allocation in Fog Computing: A Comprehensive Survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1-31, Jul. 2023. Doi: 10.1145/3586181.

[10] M. E. Manaa and M. A. Mohammed, "Data Encryption Scheme for Large Data Scale in Cloud Computing," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2, pp. 1-5, 2017.

[11] A. Gerodimos, L. Maglaras, M. A. Ferrag, N. Ayres and I. Kantzavelou, "IoT: Communication protocols and security threats," *Internet Things Cyber-Physical Syst.*, vol. 3, pp. 1-13, Jan. 2023. Doi: 10.1016/j.iotcps.2022.12.003.

[12] P. Kumar, R. Kumar, G. P. Gupta and R. Tripathi, "A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing," *Trans Emerg. Tel Tech.*, vol. 32, no. 6, pp. e4112, 2021. Doi: 10.1002/ett.4112.

[13] S. Bishnoi, S. Mohanty and B. Sahoo, "A Deep Learning-Based Methodology in Fog Environment for DDOS Attack Detection," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021.

[14] K. Kalaivani and M. Chinnadurai, "A Hybrid Deep Learning Intrusion Detection Model for Fog Computing Environment," *Intell. Autom. Soft Comput.*, vol. 30, no. 1, pp. 1-15, 2021. Doi: 10.32604/iasc.2021.017515.

[15] S. P. K. Gudla, S. K. Bhoi, S. R. Nayak and A. Verma, "DI-ADS : A Deep Intelligent Distributed Denial of Service Attack Detection Scheme for Fog-Based IoT Applications," *Math. Probl. Eng.*, vol. 2022, pp. 1-17, 2022. Doi: 10.1155/2022/3747302.

[16] H. Zhou, S. Pal, Z. Jadidi and A. Jolfaei, "A Fog-Based Security Framework for Large-Scale Industrial Internet of Things Environments," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 64-68, Mar. 2022. Doi: arXiv:2209.13323v1.

[17] R. J. Alzahrani and A. Alzahrani, "A Novel Multi Algorithm Approach to Identify Network Anomalies in the IoT Using Fog Computing and a Model to Distinguish between IoT and Non-IoT Devices," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, p. 19, Feb. 2023, Doi: 10.3390/jsan12020019. [Online]. Available: http://dx.doi.org/10.3390/jsan12020019

[18] K. Kethineni and G. Pradeepini, "Intrusion Detection in Internet of Things-Based Smart Farming Using Hybrid Deep Learning Framework," *Cluster Comput.*, pp. 1-14, Jan. 2023. https://doi.org/10.1007/s10586-023-04052-4 .

[19] A. K. Dey, G. P. Gupta and S. P. Sahu, "Hybrid Meta-Heuristic based Feature Selection

Mechanism for Cyber-Attack Detection in IoT-enabled Networks," *Procedia Comput. Sci.*, vol. 218, pp. 318-327, Jan. 2023. Doi: 10.1016/j.procs.2023.01.014.

[20] R. F. Najeeb and B. N. Dhannoon, "Improving Detection Rate of the Network Intrusion Detection System Based on Wrapper Feature Selection Approach," *Iraqi J. Sci.*, vol. 59, no. 1B, pp. 426-433, Mar. 2018. Doi: 10.24996/ijs.2018.59.1B.23.

[21] A. A. Abdualrahman and M. K. Ibrahem, "Intrusion Detection System Using Data Stream Classification," *Iraqi J. Sci.*, vol. 62, no. 1, pp. 319-328, Jan. 2021. Doi: 10.24996/ijs.2021.62.1.30.

[22] C. Krishna and V. Paul, "Entropy-Based Feature Selection using Extra Tree Classifier for IoT Security," *Iraqi J. Sci.*, vol. 64, no. 5, pp. 2466-2480, May. 2023. Doi: 10.24996/ijs.2023.64.5.31.

[23] H. A. A. Al-Khamees, N. Al-A'araji and E. S. Al-Shamery, "An Evolving Fuzzy Model to Determine an Optimal Number of Data Stream," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 22, no. 3, pp. 267-275, Sep. 2022. Doi: 10.5391/IJFIS.2022.22.3.267.

[24] H. A. A. Al-Khamees, N. Al-A'araji and E. S. Al-Shamery, "Classifying the Human Activities of Sensor Data Using Deep Neural Networks," in *International Conference on Intelligent Systems and Pattern Recognition*, Hammamet, Tunisia, Springer, Cham., vol 1589. pp 107–118 2022. https://doi.org/10.1007/978-3-031-08277-1_9

[25] B. Benmouna, R. Pourdarbani, S. Sabzi, R. Fernandez-Beltran, G. García-Mateos and J. M. Molina-Martínez, "Comparison of Classic Classifiers, Metaheuristic Algorithms and Convolutional Neural Networks in Hyperspectral Classification of Nitrogen Treatment in Tomato Leaves," *Remote Sens.*, vol. 14, no. 24, p. 6366, 2022. Doi: 10.3390/rs14246366.

[26] H. Xu, Z. Sun, Y. Cao and H. Bilal, "A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things," *Soft Comput.*, vol. 27, no. 19, pp. 14469-14481, Jul. 2023. Doi: 10.1007/s00500-023-09037-4.

[27] H. A. A. Al-Khamees, W. R. H. Al-jwaid and E. S. Al-shamery, "The impact of using Convolutional Neural Networks in COVID-19 tasks: A Survey," *Int. J. Comput. Digit. Syst.*, vol. 11, no. 1, pp. 189-197, Feb. 2022. Doi: 10.12785/ijcds/110194.

[28] H. A. A. Al-Khamees, N. Al-A'araji and E. S. Al-Shamery, "Enhancing the stability of the deep neural network using a non-constant learning rate for data stream," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 2123-2130, Apr. 2023. Doi: 10.11591/ijece.v13i2.pp2123-2130.

[29] V. Gugueoth, S. Safavat and S. Shetty, "Security of Internet of Things (IoT) using federated learning and deep learning — Recent advancements, issues and prospects," *ICT Express*, vol. 9, no. 5, pp. 941-960, Oct. 2023. Doi: 10.1016/j.icte.2023.03.006.

[30] A. A. Alshdadi, "Evaluation of IoT-Based Smart Home Assistance for Elderly People Using Robot," *Electronics*, vol. 12, no. 12, p. 2627, Jun. 2023. Doi: 10.3390/electronics12122627.

[31] I. Syamsuddin and O. M. Barukab, "SUKRY: Suricata IDS with Enhanced kNN Algorithm on Raspberry Pi for Classifying IoT Botnet Attacks," *Electronics*, vol. 11, no. 5, p. 737, Feb. 2022. Doi: 10.3390/electronics11050737.

[32] J. Li, "IOT security analysis of BDT-SVM multi- classification algorithm," *Int. J. Comput. Appl.*, vol. 45, no. 2, pp. 170-179, 2020. Doi: 10.1080/1206212X.2020.1734313.

[33] S. K. Rath, M. Sahu, S. P. Das, S. K. Bisoy and M. Sain, " A Comparative Analysis of SVM and ELM Classification on Software Reliability Prediction Model," *Electronics*, vol. 11, no. 17, p. 2707, Aug. 2022. Doi: 10.3390/electronics11172707.

[34] H. Wang, G. Li and Z. Wang, "Fast SVM classifier for large-scale classification problems," *Inf. Sci.*, vol. 642, p. 119136, Sep. 2023. Doi: 10.1016/j.ins.2023.119136.

[35] N. Potu, C. Jatoth and P. Parvataneni, "Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 23, p. e6163, 2021. Doi: 10.1002/cpe.6163.

[36] G. Goel, R. Tiwari, A. Anand and S. Kumar, "Workflow scheduling using optimization algorithm in fog computing," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021*, Singapore, Springer, vol 2, pp. 379–390, 2021, https://doi.org/10.1007/978-981-16-2597-8_32.