Journal of Autonomous Intelligence (2024) Volume 7 Issue 2
doi: 10.32629/jai.v7i2.962

ORIGINAL RESEARCH ARTICLE

IoT intrusion detection system using ensemble classifier and hyperparameter optimization using tuna search algorithm

P. M. Vijayan, S. Sundar*

School of Electronics Engineering (SENSE), Vellore Institute of Technology, Vellore 632014, Tamil Nādu, India

*** Corresponding author:** S. Sundar, sundar.s@vit.ac.in

ABSTRACT

The Internet of Things (IoT) is a dynamic and delightful research field in this emerging technology. It can be globally connected with many IoT devices and exchange a large amount of data. However, the threats also developed and misguided the entire network's behaviour. This article proposes an Intrusion Detection System (IDS) using the proposed ensemble classifier along with the Tuna Swarm Optimization (TSO) to fine-tune the hyperparameters and help to enhance the detection accuracy of attacks that take place in IoT environment. Here, the publicly available message queue telemetry transport (MQTT) network dataset is used to classify the given data into the following categories: SlowlTe, malformed, brute force, flood, DoS, and legitimate. Initially, the dataset is pre-processed to remove possible outliers, then data balancing is performed using the Synthetic Minority Oversampling Technique (SMOTE) technique and features are extracted with the help of Recursive Feature Elimination (RFE). Finally, ensemble classifier along with the optimized parameters using TSO helps in detecting the attacks in IoT attacks. The proposed TSO-ensemble classifier achieved a classification accuracy of 99.12%. In contrast, the classification accuracy of the existing Improved Vulture Starvation-based African Vultures Optimization (IVS-AVOA) and Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) have achieved a classification accuracy of 96.61% and 98.94% respectively.

Keywords: MQTT; IoT security; Internet of things; intrusion detection system; SMOTE; machine learning classifiers; TSO

ARTICLE INFO

Received: 11 July 2023
Accepted: 16 October 2023
Available online: 28 November 2023

1. Introduction

With the progress of computer and communications networks, internet technology has provided more suitable services to people around the world than ever before. Soon, this advancement of machine-to-machine (M2M) principles or the IoT will be the logical choice. The security of IoT devices has recently been a major worry, particularly in the healthcare arena[1]. Along the same lines as the growth of machines' processing capabilities, numerous techniques that are based on Machine Learning (ML) and Deep Learning (DL) have been effectively built in medical sectors as a classification and recognition platform[2]. where recent assaults have revealed catastrophic IoT security vulnerabilities[3]. Traditional network security technologies are well established. However, conventional security processes cannot be utilized directly to defend IoT devices and networks from cyber-attacks due to the resource restrictions of IoT devices and the unusual behaviour of IoT protocols[4]. As a result, IoT can be attacked in various ways depending on where the attack happens. With physical attacks, the attacker has physical access to the device and can thus damage it or physically manipulate it. IoT refers

to several protocols to ensure reliable and secure data transfer between devices[5–7]. Hence, it enables to connection of the sensors and actuators with the help of the protocols, which include Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT), and Extensible Messaging Presence Protocol (XMPP). However, MQTT is popular because it supports communication at low bandwidths, low memory, and reduced packet loss[8–10]. The MQTT, the central server, is known as the broker, and it serves as the recipient of messages from the client, which is effectively the entire node involved in the communication process[11]. Data exchanges among the nodes as the message in the form of a publish and subscribe topic[12]. The MQTT protocol uses broker/server facilities to exchange messages among the IoT nods, so it is often vulnerable to security threats. Hence it is necessary to deploy a preventive mechanism in the form of an Intrusion Detection System (IDS) to protect the IoT context[8].

Researchers and anti-malware communities have recently developed malware detection and analysis systems using machine learning and deep learning[13,14]. These systems have been separated into two different areas: 1) feature extraction and 2) feature reduction. The feature extraction process involves converting raw data into numerical features, which can be processed to maintain the original information. Feature reduction, also known as dimension reduction, reduces the number of features in a computation without sacrificing crucial information[15–17]. Many researchers have achieved promising results by applying a diverse set of algorithms, but there is considerable overlap across studies, and the collaborative use of multiple efficient tools is sluggish to emerge[18].

A Machine Learning algorithm can be used to drive, control production processes, scan for malicious files, etc.[13,19]. A machine learning system can predict the future with near-perfect accuracy without a doubt. ML has proven useful in a wide range of applications, including intrusion detection systems for IoT[20,21]. Machine learning methods not only detect but also forecast details of attacks[22–24]. As a result, this paper aims to present IDS for the MQTT protocol using machine learning approaches.

The remaining of this work is organized as follows. Section 2 provides literature survey. Section 3 provides a detail of the proposed method in our study. Section 4 results and discussion and finally, section 5 provide a conclusion and outlook for the future.

The main contribution of this paper is described as given below:

- To design the intrusion detection system framework with an ensemble classifier by tuning the hyper parameters using tuna swarm optimization (TSO) algorithm.
- To implement an ensemble classifier that combines multiple machine learning algorithms such as random forest (RF), XGBoost, LightGBM (LGBM), and CatBoost is another novel contribution.
- The paper conducts a comparative analysis with existing methods, highlighting the superior performance of the proposed TSO-ensemble classifier such as accuracy, precision, recall and F-1 score.

2. Related work

A review of an existing technique for detecting the attacks is included in this section.

Vaccari et al.[25] a new dataset called MQTT set, which focuses on IoT networks utilizing the MQTT protocol. They not only generated but also thoroughly examined this dataset by integrating hypothetical detection systems. This involved combining both cyber-attack and legitimate datasets within the MQTT network. Through experimental analysis, they demonstrated the MQTT set effectiveness in training machine learning models to implement detection systems for securing IoT environments. In 2022, Siddharthan et al.[26] proposed an IDS system for predicting cyber-attacks employing advanced elite machine learning algorithms (EML). Additionally, they adopted a streamlined protocol to address time-sensitive issues. To validate the model they created, they conducted tests using a setup that included hardware. In this setup, various sensors were interconnected using the MQTT protocol.

Vijayan and Sundar[27] have introduced an intrusion detection system using improved vulture starvation-based African vultures optimization (IVS-AVOA) for integrating the features and the hybrid fuzzy with 1DCNN as a classifier. At first, IoT data is obtained from MQTT dataset and the fed into the stage of pre-processing. After this, features are selected using IVS-AVOA and optimal autoencoder. After selection of features, the categorization is performed with the help of hybrid classifier of fuzzy and 1D-CNN. Finally, tuning of hyper-parameters takes place using AVOA. However, the rules should be periodically updated for fuzzy classification system.

Alzahrani and Akdhyani[28] developed an IDS with the help of artificial intelligent based algorithms. The MQTT protocol IoT intrusions are detected using k-nearest neighbors (KNN) algorithm, linear discriminate analysis (LDA), a convolution neural network (CNN), and a convolution neural network long short-term memory (CNN-LSTM). Among the fore mentioned algorithms, the suggested CNN-LSTM have secured better results in detecting the intrusions. However, the detection capability of the suggested approach was limited with MQTT protocols. Liu et al.[29] have introduced a particle swarm optimization (PSO) based gradient descent to detect the intrusions in IoT environment. The features are extracted using PSO and the malicious data is detected using One-Class Support Vector Machine (OCSVM). The OCSVM classifies the malicious data into two classes such as normal and abnormal data. However, the PSO based gradient descent approach was not suitable for high frequency data.

Alqahtani[30] have introduced firefly swarm optimized long short-term memory (FSO-LSTM) based intrusion detection system to predict various classes of attacks in IoT environment. The spatial and temporal correlated features were extracted using convolution neural network (CNN) and the suggested FSO-LSTM was used to predict various attacks in the IoT network. The suggested approach effectively minimized the computational complexity by performing an enhanced search using FSO and helps in detection of various classes of attacks. Han et al.[31] have introduced an intrusion detection hyperparameter control system (IDHCS) based on proximal policy optimization (PPO). The feature extraction was performed using deep neural network (DNN) and intrusions were detected using the k-means clustering approach. The PPO algorithm effectively optimize the features by analyzing the data characters in limited time. But, IDHCS was incapable to analyze the data based on real-time environment. Some of the existing approaches utilized to perform intrusion detection in IoT systems are showcased in **Table 1**.

Table 1. Features and challenges of existing Intrusion detection in IoT devices.

Methodology	Features	Challenges	Reference & year
Machine learning	Improves performance indicators including accuracy and F-1 score.	It does not modify the presented machine learning hyperparameters	[25] 2020
EML	It accomplishes greater precision.	It does not use deep learning models and optimization to develop further	[26] 2022
Fuzzy and 1D-CNN	To enhance performance and achieve precise negative predictive value, several improvements can be made.	Rules should be periodically updated	[27] 2022
CNN-LSTM	Better results in detecting the intrusions.	The detection capability of the suggested approach was limited with MQTT protocols	[28] 2022
PSO & OCSVM	Given the better results	Not suitable for high-frequency data	[29] 2021
FSO-LSTM	To predict various attacks in the IoT network	Computational complexity	[30] 2022
PPO	Effectively optimize the features	Incapable to analyze the data based on real-time environment	[31] 2022
RBFNNs	Specificity, F1, recall, precision, and accuracy	DL method cannot perform well on new sample sets when lacking data	[32] 2023
Survey paper	Detect possible intrusions into software systems	Lack of recent research articles	[33] 2022

From the literature, it was observed that the comparative study is only carried out without much attention given to a few of the factors, such as hyper parameter tuning, hybrid ML approaches, optimization algorithms, and ensemble methods. Hence, the primary novelty in our work proposes tuning the hyper parameter in the efficient machine learning algorithm in multiclass classification. such as a RF classifier, XGBoost classifier, LGBM classifier, and CatBoost classifier. This is aimed to improve the detection accuracy. Here the balanced dataset was used to train and validate the model.

3. Proposed method

The proposed intrusion detection system is intended with efficient ML classifier to detect intrusions in IoT network a with by tuning their hyperparameters. The MQTT dataset was used to detect the multiclass classification attacks to get better performance.

The data is acquired from the MQTT dataset for evaluating the detection efficiency. The obtained data undergoes the pre-processing step for data analysis and cleaning. The data obtained from the stage of pre-processing is fed into the stage of feature extraction which is used to synthesize data where the features are continuous and a classification problem and try to oversample the data using in this technique. The block diagram of proposed method is depicted in **Figure 1**.

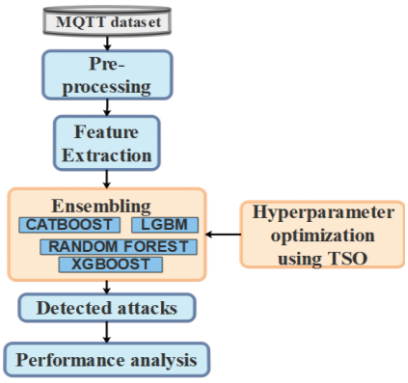

Figure 1. Diagrammatic representation of multi-class intrusion detection system in IoT.

3.1. Dataset

The scientific and industrial sectors have contributed Kaggle, an open-source database, for the experimental MQTT dataset[28]. The dataset data are collected from various IoT sensors used in home applications they are like door opening close, motion, smoke, CO-Gas, humidity, light intensity, smoke, temperature status at various period of intervals. The dataset includes malware as well as original traffic data. The size of data is 231,646 and 34 features, it includes six different types of attacks like SlowlTe, malformed, brute force, flood, DoS and legitimate. This dataset provides support for utilizing data analysis techniques or Machine Learning/Artificial Intelligence in the context of IoT. Here, the various types of attacks are explained as follows.

MQTT publish flood: attack is a type of attack where a large volume of MQTT data is transmitted over malicious IoT devices. In this attack, the objective is to overwhelm the target by flooding it with a high volume of MQTT publish messages. The MQTT publish flood attack can pose significant security risks to IoT systems that rely on MQTT as their communication protocol. It can lead to service degradation or complete unavailability, making it difficult for legitimate devices to communicate effectively.

Brute force authentication: is a type of attack where an attacker systematically tries all possible combinations of usernames and passwords to gain unauthorized access to an MQTT system. In this attack, the attacker aims to exploit weak or easily guessable credentials by attempting various combinations until they find the correct one.

Malformed: data the goal of this attack is to exploit vulnerabilities in the broker or the target service by introducing unexpected or invalid data. This can lead to service disruption, crashes, or other types of system failures. The attack takes advantage of weaknesses or flaws in the implementation of the MQTT protocol or in the processing of incoming data by the MQTT broker.

Flooding Denial-of-Service (DoS): The attacker creates numerous connections to the broker, sending a large number of MQTT requests or messages simultaneously. This flood of requests consumes network resources, processing power, and memory, making it difficult for the broker to handle legitimate client requests effectively. As a result, the services offered by the broker may be severely impacted or completely unavailable to legitimate clients.

SlowITe: the attacker establishes multiple connections to the target server or application, but instead of overwhelming it with a high volume of traffic like traditional DoS attacks, the attacker sends HTTP requests very slowly. The purpose is to exhaust the server's resources, such as open connections, available threads, or processing capacity, by keeping the connections open for as long as possible with minimal data transfer.

The following features are extrapolated and provided by the MQTT set. Such features were extracted for both legitimate and malicious cases. they are tabulated in **Table 2**.

Table 2. Features of MQTT set.

S no	Features	Characteristics
1	tcp.flags	TCP packet transfers
2	tcp.time_delta	TCP delta time measures between the prior and current packet
3	tcp.len	TCP header length
4	mqtt.conack.flags	MQTT CONNECT and response messages
5	mqtt.conack.flags.reserved	Reserved flag in the CONNECT
6	mqtt.conack.flags.sp	Session present flag in the CONNECT
7	mqtt.conack.val	Extracted the sequence data from the packet
8	mqtt.conflag.cleansess	Clean session flag
9	mqtt.conflag.passwd	Password file specified
10	mqtt.conflag.qos	Quality of service level
11	mqtt.conflag.reserved	Reserved
12	mqtt.conflag.retain	Will retain
13	mqtt.conflag.uname	MQTT, user name flag
14	mqtt.conflag.willflag	Will flag
15	mqtt.conflags	Connection flags
16	mqtt.dupflag	Duplicate flags
17	mqtt.hdrflags	Indicates header flags
18	mqtt.kalive	Keep alive MQTT
19	mqtt.len	Message length
20	mqtt.msg	Message
21	mqtt.msgid	Message ID from an incoming MQTT message
22	mqtt.msgtype	MQTT message type belonging
23	mqtt.proto_len	Protocol name and length
24	mqtt.protoname	Protocol name
25	mqtt.qos	Quality of service level (Qos 1, Qos 2, Qos 3)
26	mqtt. retain	The retain flag

Table 2. (*Continued*).

S no	Features	Characteristics
27	mqtt.sub.qos	Request for quality of service
28	mqtt.suback.qos	Granted for quality of service
29	mqtt.ver	Version of MQTT
30	mqtt.willmsg	Will message
31	mqtt.willmsg_len	Length of will message
32	mqtt.willtopic	The topic of will message
33	mqtt.willtopic_len	The topic and length of will message
34	Target	Provides the output

3.2. Pre-processing

After the stage of data acquisition from MQTT dataset, the obtained data is pre-processed by data cleaning to process it in further stages. The process of removing the improper, identical or the unfinished data in the dataset is known as data cleaning. The presence of imbalance classes in the dataset affects the performance of the classifiers. So, this research utilized Synthetic Minority Over-Sampling Technique (SMOTE) to overcome the problems related to data imbalance. SMOTE creates minority of positive samples to attain the state of class balance. Initially, the k-nearest neighbors (KNNs) y of the sample x is identified from the minority class and the new samples are generated using the random interpolation operation based on the Equation (1) as follows:

$$x_{\text{new}} = x + (y - x) \times \delta \tag{1}$$

where, the random number which lies among the interval [0, 1] is represented as δ.

3.3. Feature extraction

After the stage of pre-processing, features are extracted using RFE approach. Recursive feature elimination (RFE) is a process that begins by ranking all dataset features based on their relevance to the classification task. In the context of intrusion detection, certain features such as MQTT topic, tcp.window_size, password, MQTT client Id, username, tcp.stream, communication times, Source or destination addresses, iRTT, and ports are considered unnecessary and are subsequently removed from the dataset. This step ensures that the network data used for intrusion detection is appropriately refined. After each feature elimination, the machine learning model is re-evaluated using the reduced feature set. Performance metrics like accuracy and F-1 score are assessed to gauge if the removal of features has a detrimental effect on the model's predictive capabilities. This process of feature elimination and model re-evaluation is iteratively carried out until a specified number of features remain in the dataset or until the performance metric no longer demonstrates significant improvement.

The features mentioned in above **Table 1** are considered relevant because they have been extracted from the MQTT protocol dataset. This dataset includes various attack categories (e.g., SlowlTe, malformed, brute force, flood, DoS) as well as legitimate traffic, making it suitable for the detection task.

The RFE learning use the sequential data to learn algorithm which is represented in Equation (2) as follows:

$$x_i, y_i, i = 1, \ldots, m \tag{2}$$

where, the specified output of the feature vector x_i is represented as y_i. The sequential pair is denoted as $\{x_{i,} y_{i,}\}$ which lies among the range of (−1, 1). The optimistic hyperplane is developed based on the Equation (3) as follows:

$$f(x) = W^T x + b \tag{3}$$

where, the optimum value of the weighted vector is represented as W and the biasing b for the model is f. The class is comprised by assigning the positive and negative labels which lies in the condition $f(x) > 0$, for the feature x. The feature x fulfils condition which is defined in Equation (4) as follows:

$$W^T x + b = 0 \tag{4}$$

Distance among training data is represented in the form of vectors. The objective function defined in Equation (5) is used to increase and decrease the distance among the vectors.

$$\text{MaxMin}_{w,b} = |x - x_i| : W^T x = b = 0, i = 1, \ldots, m \tag{5}$$

where W is normal to hyperplane and $|b||w|$ is referred as perpendicular distance between hyperplane and origin.

3.4. Classification using ensemble learning

An ensemble learning acts as an important machine learning technique which is used for an effective classification. Ensemble classifiers combine the predictions of multiple base classifiers or models. By doing so, they leverage the diversity of these models to enhance overall accuracy and robustness. In the context of IoT intrusion detection, where the threat landscape is diverse and rapidly evolving, having a more accurate and robust system is crucial. Ensemble methods have been proven to reduce the risk of overfitting and improve generalization compared to individual models. This is a reason why ensemble methods are often better than individual machine learning models. In this research, ensemble is performed among four classifiers of same family. The four classifies include CatBoost, LGBM, RF and XGBoost algorithms.

CatBoost: It is a ML algorithm which is related to family of gradient boosting decision tree (GBDT). This algorithm is fine for heterogenous tasks. Moreover, the CatBoost algorithm prevents the overfitting using the overfitting detector, that utilize the initial parameters to control the count of trees to be created.

LGBM: It is a type of gradient boosting framework which utilize tree-based learning algorithms. LGBM implies on two sampling approaches known as Gradient-based One Side Sampling (GOSS) and exclusive feature bundling (EFB). GOSS eliminate the data instance with small gradients and helps to evaluate the information gain. EFB groups the mutually exclusive features by reducing the number of irrelevant features. Training the model using GBDT enhance the accuracy of predicting attacks.

XGBoost: It is utilized to perform predictions for an unorganized data using DT algorithm. XGBoost has built in support for parallel processing which helps to train the model with large dataset with minimal time period. Moreover, XGBoost is known for its scalability which can work can work on any vulnerable conditions.

RF: It is a type of ensemble algorithm which utilize ensemble learning approach to categorize and predict the type of attacks. RF is developed using the bootstrap samples which is used to train the data with a randomized feature selection approach. RF is considered as one of the most versatile tools to predict the accuracy for classifying the attacks.

In this research, ensemble is performed using weighted majority voting (WMV) approach. The performance of the classifiers is more qualified than others. In WMV, every individual vote is weighted by the prediction accuracy of the classifier, the classifier which exhibits better accuracy is considered for the voting process. The total votes of the classifier are evaluated using the Equation (6) as follows:

$$T_v = \sum_{l=1}^{M} Acc(A_l) \times F_k(c_l) \tag{6}$$

where, the prediction accuracy of the classifier is denoted as Acc and the count of votes is represented as c_l. Thus, the classifier which has greatest total weight is chosen for the process of classification.

Hyperparameter optimization using TSO

The output from the ensemble model is fed into the process of hyperparameter optimization which helps to improve the detection accuracy of the model. The optimization of hyperparameters is a significant stage to regulate the behaviour of the ensemble model. The improper optimization of hyperparameters leads improve

the loss function and provides improper results. So, the hyperparameter optimization is performed to obtain a better classification result. This research introduced a swarm-based optimization technique known as tuna swarm optimization (TSO) algorithm to optimize the parameters and the functionalities. The TSO algorithm has global level search ability and higher exploration efficiency which helps to fine tune the hyper-parameters in an effective manner. The iterative process involved in TSO is described as follows:

The tuna is a carnivorous fish which lives in the marine surface. A single tuna can swim faster but their efficiency not as fast as nimble fish. So, tunas perform group travel method for predation. These creatures are known for its effective and intelligent foraging strategies to detect and attack their prey. Tuna performs two kind of strategies such as spiral foraging and parabolic foraging.

TSO enhances ensemble classifier performance by optimizing hyperparameters in the IDS. Hyperparameters significantly affect machine learning model performance, and TSO fine-tunes them, maximizing accuracy. TSO adeptly navigates the hyperparameter space, reducing the likelihood of overlooking optimal settings. It balances exploration and exploitation, yielding improved model configurations. With its global search capability, TSO increases the likelihood of discovering global optima for the ensemble classifier. The outcome is a highly accurate intrusion detection system that precisely identifies IoT security threats. Some unique features are mentioned below: global search ability, higher exploration efficiency**,** dynamic coefficients, parallel processing, minimization of computational complexity.

Initialization: it is the foremost stage in most of the optimization techniques. In TSO, the initial populations are created in a randomized manner in a uniform search space which is mathematically expressed in Equation (7) as follows:

$$X_i^{\text{int}} = rand \times (u_b - l_b) + lb, i = 1, 2, \dots, NP \tag{7}$$

where, X_i^{int} is the individual at the initial stage, the upper and the lower bounds of search space is represented as u_b and l_b respectively. The total population of tuna is represented as NP. The randomized vector which is distributed in a uniform space is denoted as $rand$ and it lies among the range of 0 to 1.

Spiral foraging: when the school of small fishes change their direction to safeguard them from the predators, it becomes complex for the predators to lock their prey. Tuna chases its prey by creating a spiral formation. After spiraling their target, tuna transfer the information to their neighbor's. This strategy of tuna is mathematically presented in Equation (8) as follows:

$$X_i^{t+1} = \begin{cases} \alpha_1 \times (X_{\text{best}}^t + \beta \times \left|X_{\text{best}}^t - X_i^t\right| + \alpha_2 \times X_i^t, & i = 1 \\ \alpha_1 \times (X_{\text{best}}^t + \beta \times \left|X_{\text{best}}^t - X_i^t\right| + \alpha_2 \times X_{i-1}^t, & i = 2, 3, \dots, NP \end{cases} \tag{8}$$

where, the value of $\alpha_1, \alpha_2, \beta$ and l is computed using the Equations (9)–(12) as follows:

$$\alpha_1 = a + (1 - a) \times \frac{t}{t_{\max}} \tag{9}$$

$$\alpha_2 = (1 - a) - (1 - a) \times \frac{t}{t_{\max}} \tag{10}$$

$$\beta = e^{bl} \times \cos(2\pi b) \tag{11}$$

$$l = e^{3\cos\left(\left((t_{\max} + 1/t) - 1\right)\pi\right)} \tag{12}$$

where, the optimistic individual at the current position is represented as X_{best}^t. The weighted co-efficient that regulate the individual at the optimal state and the previous state is represented as α_1 and α_2. The constant which is utilized to distinguish operation of tuna which track the optimistic individual is represented as a. The iteration at the current state and maximum iterations is represented as t and $t_{\max}$ respectively. The random number which is distributed among the range of 0 and 1 is represented as b.

The exploitation ability of the tuna is improved while they forage in a spiral manner. But, the probability of finding food for every tuna in the school is improbable. So, a randomized coordinate point is created in a spiral search which helps the individuals to perform a wider search which is described in Equation (13) as follows:

$$X_i^{t+1} = \begin{cases} \alpha_1 \times (X_{rand}^t + \beta \times \left|X_{rand}^t - X_i^t\right| + \alpha_2 \times X_i^t, i = 1 \\ \alpha_1 \times (X_{rand}^t + \beta \times \left|X_{rand}^t - X_i^t\right| + \alpha_2 \times X_{i-1}^t, i = 2, 3, \ldots, NP \end{cases} \tag{13}$$

where, the randomly generated reference point is represented as X_{rand}^t.

Thus, TSO vary reference points of spiral foraging from random to the optimal values when iteration gets increased. Finalized expression related to spiral foraging strategy is represented in Equation (14) as follows:

$$X_i^{t+1} = \begin{cases} \alpha_1 \times (X_{best}^t + \beta \times \left|X_{best}^t - X_i^t\right| + \alpha_2 \times X_i^t, \quad i = 1 \\ \alpha_1 \times (X_{best}^t + \beta \times \left|X_{best}^t - X_i^t\right| + \alpha_2 \times X_{i-1}^t, \quad i = 2, 3, \ldots, NP, \text{if } rand < \dfrac{t}{t_{\max}} \\ \alpha_1 \times (X_{rand}^t + \beta \times \left|X_{rand}^t - X_i^t\right| + \alpha_2 \times X_i^t, i = 1 \\ \alpha_1 \times (X_{rand}^t + \beta \times \left|X_{rand}^t - X_i^t\right| + \alpha_2 \times X_{i-1}^t, i = 2, 3, \ldots, NP, \text{if } rand \geq \dfrac{t}{t_{\max}} \end{cases} \tag{14}$$

Parabolic foraging: tuna performs parabolic development and hunt for the food source by searching among themselves. The two-process performed by tuna with selection probability of 50%. This act is mathematically represented in Equation (15) as follows:

$$X_i^{t+1} = \begin{cases} X_{\text{best}}^t + rand \times \left(X_{\text{best}}^t - X_i^t\right) + TF \times p^2 \times \left(X_{\text{best}}^t - X_i^t\right), \text{if } rand < 0.5 \\ TF \times p^2 \times X_i^t, \text{if } rand \geq 0.5 \end{cases} \tag{15}$$

where, the value of $\rho = \left(1 - \frac{t}{t_{\max}}\right)^{(t/t_{\max})}$ and the random number which lies in the range of 1 or −1 is denoted as TF. The each individual randomly selects two foraging techniques to recreate a position in the search space based on probability z. At the time of optimization process, the individuals are updated and evaluated till the optimal value is obtained.

4. Results and analysis

This section provides a detailed discussion of the results obtained from the proposed approach. The proposed TSO-ensemble classifier is implemented using python as the implementation platform and the system used has the specification of Intel i7 processor, 8 GB RAM and Windows 11 operating system. Moreover, performance of proposed approach is evaluated by means of accuracy, sensitivity, F-1 score and precision which is presented in Equations (16)–(19) as follows:

Accuracy

Accuracy simply calculates how often the classifier guesses accurately. The ratio between the number of right forecasts to the total number of predictions is the solution of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{16}$$

Precision

Many of the cases predicted to be positive were correct. False positives are more of a concern than false negatives, therefore precision is useful.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{17}$$

Recall (sensitivity)

The model's ability to accurately predict good outcomes is shown in the number of recalls. When a false negative is more worrisome than a false positive, this is a useful indicator.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

F-measure

It provides an overview of the precision and recalls measures. It is greatest when precision equals recall. The F-measure is measured using the mean of precision and recall.

$$\text{F} - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{19}$$

4.1. Performance analysis

In this performance of the existing classifiers are evaluated based on two cases. In first case, the performance of the classifier is evaluated with actual parameters and in second case, the performance of the classifier is evaluated with optimized parameters based on accuracy, precision, recall and F-measure. The **Table 3** depicted below presents the performance of the classifier for actual parameters.

Table 3. Performance analysis of the classifier for the actual parameters.

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-1 score (%)
RF	92.84	91.44	90.23	91.73
XG-Boost	93.11	94.10	90.98	90.33
LGBM	93.45	93.75	92.64	92.18
CatBoost	95.76	95.64	91.49	91.74
Ensemble	97.84	96.23	93.45	94.48

The results from the **Table 3** shows that the ensemble classifier used in this research exhibits better result in overall metrics. For instance, the ensemble classifier used in this research achieved better classification accuracy of 97.84% whereas the existing classifiers such as RF, Extra Tree and CatBoost have achieved classification accuracy of 92.84%, 93.11%, 93.45% and 95.76%, respectively. The better result of the ensemble classifier is due to the prediction of multiple models and increase the overall prediction accuracy. Secondly, the performance of the classifier is evaluated with optimized parameters. The **Table 4** below presents the performance of the classifiers when evaluated with these optimized parameters.

Table 4. Performance analysis of the classifier for the optimized parameters.

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F-1 score (%)
RF	93.45	93.02	91.98	93.33
XG-Boost	94.13	96.54	92.91	92.11
LGBM	94.22	94.49	94.78	94.09
CatBoost	96.43	96.88	93.67	92.99
Ensemble	99.12	97.89	95.24	96.37

The results from the **Table 4** and **Figure 2** show that the ensemble classifier with the optimized hyperparameters using TSO have achieved better classification accuracy of 99.12% whereas the existing classifiers such as RF, XG-Boost, LGBM and CatBoost have achieved accuracy of 93.45%, 94.13%, 94.22% and 96.43%, respectively. This shows that ensemble classifier with the optimized hyperparameters using TSO have achieved better results than existing classification approaches. The hyperparameter optimization using TSO helps to achieve global level search ability and higher exploration efficiency which helps to fine tune the hyper-parameters in an effective manner.

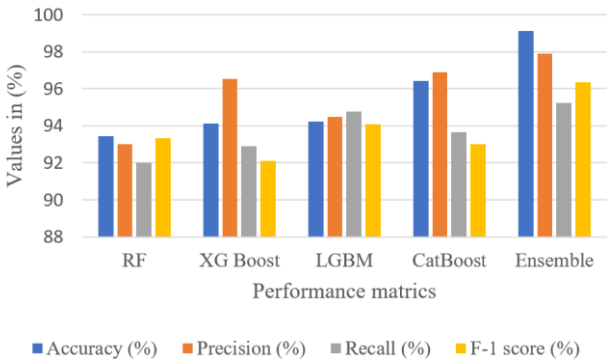

Figure 2. Graphical representation for performance of classifiers for optimized hyperparameters.

4.2. Comparative analysis

In this section, the performance of the TSO-ensemble classifier is evaluated with the existing classification approaches for detecting the attacks. The performance of the proposed approach is evaluated with existing IVS-AVOA[23] and CNN-LSTM[24] based on accuracy, precision, recall and F-1 score. The **Table 5** depicted below presents the evaluation of comparative results obtained while evaluating the proposed method with existing techniques.

This ensemble strategy allows TSO to leverage the diversity of different classifiers, making it more robust and capable of achieving better accuracy in IoT intrusion detection compared to using IVS-AVOA and CNN-LSTM in isolation. TSO has strengths in hyperparameter optimization, global search ability, and the balance between exploration and exploitation. The choice between the two approaches depends on the specific requirements and constraints of the IoT environment under consideration.

The IVS AVOA fine-tunes the hyperparameters but does not focus on ensemble techniques, similar to CNN-LSTM, which also does not primarily focus on optimization or ensemble methods.

Table 5. Comparative table.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-1 score (%)
IVS-AVOA[27]	96.61	95.28	100	95.27
CNN-LSTM[28]	98.94	95.68	96.12	95
TSO-ensemble	99.12	97.89	95.24	96.37

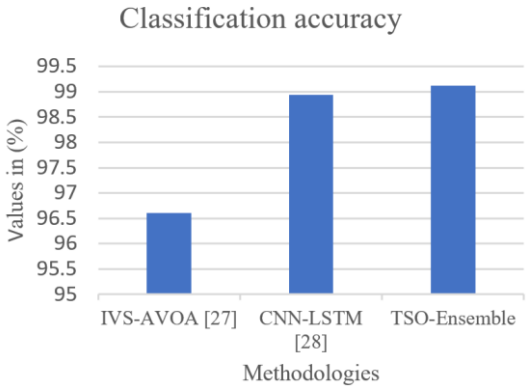

Figure 3. Graphical representation for classification accuracy.

The results acquired from **Table 3** shows that the proposed TSO-ensemble classifier have achieved better classification accuracy of 99.12% whereas the existing IVS-AVOA and CNN-LSTM have achieved the classification accuracy of 96.61% and 98.94%, respectively. The accuracy of the ensemble classifier is

improvised due to hyperparameters which is optimized using the TSO. The graphical representation for the classification accuracy for the proposed approach with existing approach is depicted in **Figure 3**.

4.3. Computational complexity

The computational requirements of the proposed system, including the ensemble classifier and hyperparameter optimization using tuna swarm optimization (TSO), need to be evaluated. IoT devices often have limited processing power and memory. Researchers should assess whether the proposed system can operate efficiently within these constraints. The computation complexities of proposed model and resource requirements specification are mentioned in **Table 6** and **Table 7**, respectively.

Table 6. The computational complexity of the state-of-the-art IDS methods.

Measures	IVS-AVOA-HC	CNN-LSTM	TSO-ensemble
Time (in a sec)	62.6778	63.7826	61.9844

Resource requirements: IoT devices typically have limited resources, such as CPU, memory, and energy. The intrusion detection system should be designed to minimize resource consumption to ensure it can run on resource-constrained devices without causing performance degradation.

Table 7. Resource requirements.

OS name	Microsoft Windows 11 Pro
Version	10.0.22000 Build 22000
System type	x64-based PC
Processor	Intel (R) Core (TM) i7-1005G1 CPU@1.20GHz, 1190 MHZ, 2 Core(s), 4 logical processor(s)
Hardware	Abstraction layer version = 10.0.22000.527
User name	DESKTOP-4DN2A9L\hi
Installed physical memory (RAM)	8.00 GB
Total physical memory	3.77 GB

Scalability: IoT environments can vary in scale, from a small network of devices to large-scale deployments. Researchers should investigate whether the proposed system can scale effectively to monitor and protect IoT networks of different sizes.

5. Conclusion

This research introduced IoT based IDS using efficient machine learning algorithms like RF classifier, Extra Tree classifier, LGBM classifier, and CatBoost classifier. The raw data is obtained from MQTT dataset and it is pre-processed using data cleaning method, then the features from the pre-processed data extracted using RFE. After this the proposed ensemble classifier with hyperparameter optimization using TSO used to improve accuracy of detecting the attacks. The results from the proposed approach shows that it has achieved better classification accuracy of 99.12% whereas the existing IVS-AVOA and CNN-LSTM have obtained classification accuracy of 96.91% and 98.94%, respectively. Thus, the obtained results prove the efficiency of the IDS framework which is better than existing methodologies. In the future, multi-modal traffic classifier behaviour can be enhanced modal traffic by explainable artificial intelligence (XAI) techniques based on deep learning.

Data availability statement

MQTTset dataset available in the following link: https://www.kaggle.com/datasets/cnrieiit/mqttset.[28]

Author contributions

Conceptualization, PMV and SS; methodology, PMV and SS; validation, PMV and SS; resources, PMV and SS; writing—original draft preparation, PMV and SS; writing—review and editing, PMV and SS. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Li R, Li Q, Zhou J, Jiang Y. ADRIoT: An edge-assisted anomaly detection framework against IoT-based network attacks. *IEEE Internet of Things Journal* 2022; 9(13): 10576–10587. doi: 10.1109/jiot.2021.3122148
2. Aminizadeh S, Heidari A, Toumaj S, et al. The applications of machine learning techniques in medical data processing based on distributed computing and the internet of things. *Computer Methods and Programs in Biomedicine* 2023; 241: 107745. doi: 10.1016/j.cmpb.2023.107745
3. Khan AR, Kashif M, Jhaveri RH, et al. Deep learning for intrusion detection and security of internet of things (IoT): Current analysis, challenges, and possible solutions. *Security and Communication Networks* 2022; 2022: 1–13. doi: 10.1155/2022/4016073
4. Abbas A, Khan MA, Latif S, et al. A new ensemble-based intrusion detection system for internet of things. *Arabian Journal for Science and Engineering* 2021; 47(2): 1805–1819. doi: 10.1007/s13369-021-06086-5
5. Krishna ESP, Thangavelu A. Attack detection in IoT devices using hybrid metaheuristic lion optimization algorithm and firefly optimization algorithm. *International Journal of System Assurance Engineering and Management* 2021. doi: 10.1007/s13198-021-01150-7
6. Babu MR, Veena KN. Implementing optimized classifier for distributed attack detection and BAIT-based attack correction in IoT. *International Journal of System Assurance Engineering and Management* 2021. doi: 10.1007/s13198-021-01115-w
7. Bedi P, Mewada S, Vatti RA, et al. RETRACTED: Detection of attacks in IoT sensors networks using machine learning algorithm. *Microprocessors and Microsystems* 2021; 82: 103814. doi: 10.1016/j.micpro.2020.103814
8. Bhosale SA, Sonavane SS. Wormhole attack detection system for IoT network: A hybrid approach. *Wireless Personal Communications* 2021; 124(2): 1081–1108. doi: 10.1007/s11277-021-09395-y
9. Mishra B, Kertesz A. The use of MQTT in M2M and IoT systems: A survey. *IEEE Access* 2020; 8: 201071–201086. doi: 10.1109/access.2020.3035849
10. Selvi M, Gayathri A, Santhosh Kumar SVN, Kannan A. Energy efficient and secured MQTT protocol using IoT. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 2020; 9(4): 11–14. doi: 10.35940/ijitee.b6264.029420
11. Casteur G, Aubaret A, Blondeau B, et al. Fuzzing attacks for vulnerability discovery within MQTT protocol. In: Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC); 15–19 June 2020; Limassol, Cyprus. pp. 420–425.
12. Makhija J, Shetty AA, Bangera A. Classification of attacks on MQTT-based IoT system using machine learning techniques. In: Khanna A, Gupta D, Bhattacharyya S, et al. (editors). *International Conference on Innovative Computing and Communications, Advances in Intelligent Systems and Computing*. Springer; 2021. pp. 217–224.
13. da Costa KAP, Papa JP, Lisboa CO, et al. Internet of things: A survey on machine learning-based intrusion detection approaches. *Computer Networks* 2019; 151: 147–157. doi: 10.1016/j.comnet.2019.01.023
14. Buccafurri F, De Angelis V, Nardone R. Securing MQTT by blockchain-based OTP authentication. *Sensors* 2020; 20(7): 2002. doi: 10.3390/s20072002
15. Vijayan P M, Sundar S. An Efficient CatBoost Classifier Approach to Detect Intrusions in MQTT Protocol for Internet of Things. In: Chaki N, Devarakonda N, Cortesi A (editors). *Lecture Notes on Data Engineering and Communications Technologies*, Proceedings of International Conference on Computational Intelligence and Data Engineering; Singapore. Springer; 2023. Volume V163, pp. 255–267.
16. Verma A, Ranga V. Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications* 2019; 111(4) :2287–2310. doi: 10.1007/s11277-019-06986-8
17. Kumar V, Das AK, Sinha D. UIDS: A unified intrusion detection system for IoT environment. *Evolutionary Intelligence* 2019; 14(1): 47–59. doi: 10.1007/s12065-019-00291-w
18. Amiri Z, Heidari A, Navimipour NJ, et al. Adventures in data analysis: A systematic review of deep learning techniques for pattern recognition in cyber-physical-social systems. *Multimedia Tools and Applications* 2023. doi: 10.1007/s11042-023-16382-x
19. Jeyaselvi M, Dhanaraj RK, Sathya M, et al. A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks. *Cluster Computing* 2022; 26(1): 559–574. doi: 10.1007/s10586-022-03607-1

20. Awajan A. A novel deep learning-based intrusion detection system for IOT networks. *Computers* 2023; 12(2): 34. doi: 10.3390/computers12020034
21. Zhong M, Zhou Y, Chen G. Sequential model-based intrusion detection system for IoT servers using deep learning methods. *Sensors* 2021; 21(4): 1113. doi: 10.3390/s21041113
22. Fatani A, Dahou A, Al-qaness MAA, et al. Advanced feature extraction and selection approach using deep learning and aquila optimizer for IoT intrusion detection system. *Sensors* 2021; 22(1): 140. doi: 10.3390/s22010140
23. Le KH, Nguyen MH, Tran TD, Tran ND. IMIDS: An intelligent intrusion detection system against cyber threats in IoT. *Electronics* 2022; 11(4): 524. doi: 10.3390/electronics11040524
24. Fatani A, Abd Elaziz M, Dahou A, et al. IoT intrusion detection system using deep learning and enhanced transient search optimization. *IEEE Access* 2021; 9: 123448–123464. doi: 10.1109/access.2021.3109081
25. Vaccari I, Chiola G, Aiello M, et al. MQTTset, a new dataset for machine learning techniques on MQTT. *Sensors* 2020; 20(22): 6578. doi: 10.3390/s20226578
26. Siddharthan H, Deepa T, Chandhar P. SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features. *IEEE Access* 2022; 10: 33095–33110. doi: 10.1109/access.2022.3161566
27. Vijayan PM, Sundar S. Hybrid MQTTNet: An intrusion detection system using heuristic-based optimal feature integration and hybrid fuzzy with 1DCNN. *Cybernetics and Systems* 2022; 2022: 1–34. doi: 10.1080/01969722.2022.2145649
28. Alzahrani A, Aldhyani THH. Artificial intelligence algorithms for detecting and classifying MQTT protocol internet of things attacks. *Electronics* 2022; 11(22): 3837. doi: 10.3390/electronics11223837
29. Liu J, Yang D, Lian M, Li M. Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access* 2021; 9: 38254–38268. doi: 10.1109/access.2021.3063671
30. Alqahtani AS. FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks. *The Journal of Supercomputing* 2022; 78(7): 9438–9455. doi: 10.1007/s11227-021-04285-3
31. Han H, Kim H, Kim Y. An efficient hyperparameter control method for a network intrusion detection system based on proximal policy optimization. *Symmetry* 2022; 14(1): 161. doi: 10.3390/sym14010161
32. Heidari A, Jafari Navimipour N, Unal M. A secure intrusion detection platform using blockchain and radial basis function neural networks for Internet of drones. *IEEE Internet Things Journal*. 2023; 10(10): 8445–8454. doi: 10.1109/jiot.2023.3237661
33. Heidari A, Jabraeil Jamali MA. Internet of things intrusion detection systems: A comprehensive review and future directions. *Cluster Computing* 2022; 26(6): 3753–3780. doi: 10.1007/s10586-022-03776-z