Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

Classification and detection of malware attack using hyperparameter optimization framework in SIoT

Anciline Jenifer J[1], Dr. Piramu Preethika S.K[2]

[1]Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS)
[2]Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS)

Abstract

Due to rapid advancement in World Wide Web and increased usage of internet, there is an extensive growth in Internet of Things (IoT). The IoT makes possible the operations smooth by enabling constant connectivity, granting internet access to all computing devices. Integrating and managing numerous devices is challenging with traditional methods. Nevertheless, social networks make it easier to establish communication between people. The Social Internet of Things (SIoT) enables objects to establish social connections based on human preferences. Diverse IoT devices interact to establish relationships by considering shared type of devices, attributes, and features. As Social Networks (SN) hold more and more data, they are more vulnerable to attacks by malware. As a result, malware detection is becoming a critical concern in the Internet of Things. Security can be significantly impacted by sociality in a number of ways. Advanced malware attacks must be identified using a method that is quick, dependable and efficient. This paper involves machine learning-based malware detection which looks threat identification at cyber security level and protection in the SIoT. The most advanced hyperparameter optimization framework (OPTUNA), which is used to detect malware from the Message Queuing Telemetry Transport (MQTT) dataset, was employed in this study. Furthermore, to regulate independent OPTUNA-based sampling algorithms, the proposed paper employs the Tree-Structured Parzen Estimator (TPE), which may be used to find quantization settings that maximize accuracy while minimizing latency. Moreover, the proposed OPTUNA based TPE sampler for LGBM model has accomplished the best Micro F1 score(accuracy) as 0.83 than the other classifier for identifying the types of malware attacks in SIoT.

Keywords: Internet of Things (IoT), Social Internet of Things (SIoT), OPTUNA framework, Tree-Structured Parzen Estimator (TPE), Malware detection.

Introduction

The rapid advancements in IoT are predicted to lead to a significant increase in the number of connected devices. It is anticipated that the exponential growth of these interconnected devices would result in an impressive 41 billion gadgets being incorporated into the Internet of Things by 2025 [1]. There are several scalability challenges facing the Internet of Things. In this the most important issue is the scalability explosion. The sensing layer [2] experiences a surge in data, the network layer [3] sees a proliferation of connections, and the applications layer [4] witnesses a growth in services within the IoT. IoT device characteristics have evolved over time where everything is autonomous. IoT refers to an alternative reality where objects like sensors, smartphones, and actuators have individual identities. Although they differ in terms of operating systems, platforms, communication protocols, and associated standards, among other things, they ignore these distinctions when communicating with one another. To satisfy the needs of its users, each device must establish communication with other objects in its immediate environment.

The objects in the upcoming generation of IoT are socially interconnected, making them intelligent and interactive [5]. Through the integration of physical devices with their social dimension, they can now understand the social context of their users and perform entirely novel computing tasks [6]. However, a new class of SN application that may function at the Unit IoT level has emerged as an outcome of online SN. Consequently, there has been a rapid growth in the social links between IoT entities, including relationships between users and devices. These partnerships are enabling IoT applications with essential features including social recommendation services, community administration, etc. But the growing quantity of these ties, together with their diverse social characteristics, has created a bottleneck that keeps the IoT from keeping the relationship to enhance the services it offers and personalize the material it delivers.

The integration of social media and the Internet of Things has led to the creation of a novel paradigm called SIoT. This model creates an ecosystem that makes it possible for users and tools to communicate. SIoT offers a distinct advantage over traditional IoT by the interaction among devices to facilitate distributed and autonomous learning about each other. In the SIoT, object is used to refer to each device, and each object has the capability to influence other objects. In the context of IoT, SIoT are crucial in creating an interaction between people and social objects with a network of objects [7]. Smart objects enable digitization of applications, making processing simple and effective. Data aggregation is used to organize and deduce the information gathered from social media platforms.

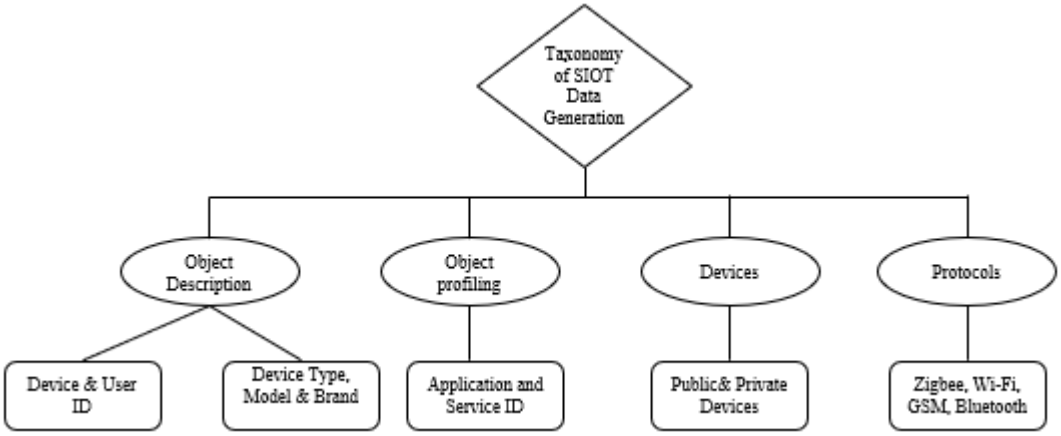

Figure.1 Taxonomy of SIoT data creation

The taxonomy of SIoT data creation is shown in Figure 1. It includes device types, object profiles, and object descriptions. Every component includes the brand and model of the device as well as private and public, stationary and mobile devices. Data within the SIoT framework are generated by devices falling into these specified groups. Data aggregation entails gathering data from the devices, combining it, and transmitting it to the base station. Several approaches are employed for data aggregation, including in-network aggregation, cluster-based aggregation, and tree-based approaches. In the cluster-based approach, the area of interest is segmented into numerous clusters. A cluster head is chosen inside each cluster for data aggregation process. Instead of transmitting the data directly to the base station, each individual device detecting the data sends it to the cluster head within the same cluster. This leads to significant energy savings in a network. The benefits of data aggregation in a cluster-based approach include reduced traffic load, energy conservation, robustness, correctness of information, and less redundancy.

IoT devices have many advantages, but there are drawbacks as well, including security, privacy, and data management issues when they are widely used. There is a greater chance of cyberattacks, data breaches, and illegal access as the number of connected devices rises. The security of IoT devices and the information produced by them must thus be given top priority, which calls for the creation of strong security mechanisms, protocols, and best practices. Furthermore, managing attacks by malware is among the primary obstacle in the IoT domain. Malicious programs or code that can harm or interfere with the operation of endpoint devices is referred to as

malware [8]. Devices infected with malware may be utilized by unauthorized users or for unintended purposes. Malware frequently takes the form of worms, ransomware, adware, spyware and Trojan horses [9].

Furthermore, because these devices have little amounts of RAM, storage, processing power, and security update capability, they are especially susceptible to malware attacks [10]. In order to address the expansion of social relationships, SIoT tools are required to communicate the social characteristics of their users with both internal and external applications. This allows the services to be customized based on the social context of the users. However, the sensitive nature of users' exposes them to vulnerability in the event of leaking data possibly resulting in socially engineered attacks on the SIoT network. With the extensive array of interconnected SIoT devices levels, data sharing rights becomes quite difficult. As a result, developing reliable and precise techniques is essential to reducing the negative effects of such attacks. While network systems employ machine learning (ML) techniques to detect and prevent malware, creating an efficient ML model that can correctly identify and stop IoT malware remains a difficult task.

In this research, a Lazy Classifier (LC) instance is generated to predict outcomes for each observation by aggregating predictions from multiple models. The framework is required to align the data for each model, employ metrics to assess the highest accuracy among models for the current dataset, and ultimately select the optimal model. The effectiveness of any classification algorithm is largely dependent on its ideal hyperparameters. The best set of hyperparameters can be chosen to increase the classification algorithm's accuracy.

To determine the ideal hyperparameter values for the ML model, an advanced hyperparameter optimisation framework [11] was used in this investigation for identifying the types of malware attacks in SIoT. As a result, among the available hyperparameters, the best appropriate set was chosen for this investigation. The OPTUNA framework modifies the hyperparameters based on ongoing learning from prior optimisations. Additionally, this research work make use of TPE technique which can handle independent sampling techniques and leads to a faster and more effective convergence.

Primary contributions of this research work is outlined as follows:

- To develop machine learning models for the classification problem and subsequently train them utilizing the pre-processed dataset.
- To assess machine learning models using different performance measures for the purpose of identifying the most suitable model, employing LP classifier.
- To optimize the hyperparameters using OPTUNA based TPE sampler to fine tune the model and validate that the suggested configuration enhances accuracy performance.
- To compare selected models and conclude on the most robust classification model to accurately detect the type of malware attack in SIoT based on the performance metrics.

Literature Review

This section presents an overview of the studies conducted by various experts in the domain of SIoT, IoT in malware detection, and various optimization models are presented.

Maran et al [12] emphasizes the security aspects of IoT sensors or devices. Machine learning (ML) algorithms are employed for the identification and exploration of malware within a dataset derived from an (IoT) device. The paper determines the superior algorithm that demonstrates greater success in malware detection within the dataset and evaluates the associated accuracy. The Random Forest algorithm yielded optimal results, yielding a 96% accuracy score, demonstrating superior results. Ali Mehrban and Pegah Ahadian [13] presented a hybrid CNN-LSTM model for the identification of IoT malware, and its performance was systematically assessed in comparison to established methods. The utilization of the CNN algorithm facilitated the construction of an advanced learning model, while the LSTM classifier demonstrated elevated accuracy in the classification process. This study provides a foundation for enhancing security measures in IoT ecosystems to make them more resilient.

Lee et al. [14] study aims to assess the feasibility of detecting IoT malware in comparison to benign entities, and to explore the classification of different malware family types. These incorporate low-dimensional and fixed length features into machine learning models by utilising opcode category information. The performance results classification and malware detection achieve an accuracy exceeding 98.0%. The results obtained from experiments demonstrate the performance and resilience of the presented features in accurately distinguishing various categories of IoT malware from benign entities. Jeon et al. [15] introduced a dynamic analysis methodology for detecting malware in IoT. The system calls, network activity, virtual file system (VFS), process and memory are among the essential components of this method. Rey et al. [16] leveraged network traffic packets from malware-infected IoT devices as input features in their analysis. Researchers are exploring the correlation or relationship among IoT malware, as IoT malware recurrently utilizes shared functions. Consequently, it has the capability to reconstruct family lineage and trace evolution processes [17]. Dhelim et al [18] explore the implications of IoT on addressing the challenge of social relationships proliferation, managing social relationships and examine proposed solutions employing ASI. Khelloufi et al [19] introduced a mechanism that utilizes social linkages among owners of IoT devices. This recommendation relies on the various connections among the service provider and requester. They also introduced a boundary-based technique for community detection, which is used to create communities of socially connected devices. Abdelghani et al. [20] emphasize the significance of trust management in SIoT, highlighting it as a crucial factor establishing a trustworthy and secure data exchange, particularly in the context of Quality of Service (QoS) for offered services. Consequently, they contribute a comprehensive survey article addressing trust models, types, associated properties, challenges, requirements and SIoT limitations. A secure relationship emphasizes two essential components as trustee and trustor, created on the basis of mutual benefits and influenced by various framework like time and location. Roopa et al. [21] discussed an extensive systematic review in accordance with SIoT, examining the contemporary articles to address few key aspects in SIoT such as object relationships and trustworthiness. The objectives were to enhance link connections among objects in networks while enumerating the fundamental prerequisites of the SIoT network. Elshewey [22] presents the hyOPTGB model, utilizing an optimized gradient boosting (GB) classifier for HCV disease prediction. By optimizing the hyperparameters with the OPTUNA framework, the accuracy of the model is improved. In comparison to alternative machine learning designs, the hyOPTGB model demonstrated high performance, attaining a 95.3% accuracy rate. Watanabe [23] explores the identification of roles played by individual parameter and examining the influences on optimization of hyperparameters. This paper elucidates the utilization of TPE, a Bayesian optimization methodology, extensively employed in contemporary parameter tuning frameworks. The outcomes illustrate the high performance of our TPE in comparison to prevailing TPE-relevance packages, including Hyperopt and Optuna. It can achieve better performance than the most advanced BO techniques with far less iterational work. An adaptation of the Tree-structured Parzen Estimator (TPE), known as c-TPE for constrained optimization, is presented by Watanabe and Hutter [24]. This addition is made possible by the simple factorization of AFs. The investigation shows that c-TPE remains powerful across different constrain stages, displaying the most superior average rank performance compared to existing approaches, with statistically notable outcomes observed in 81 settings that encompass search spaces featuring categorical parameters.

Dataset Description

The dataset is composed by 8 MQTT sensors with different features. Each component of a real network is defined in the dataset, which is made up of IoT sensors based on MQTT. In the IoT context sensors in a smart home environment gather data on temperature, light, humidity, CO-Gas, motion, smoke, doors, and fans over a range of time intervals because each sensor behaves differently from the others. Initially the dataset contains a total of 330926 records, and it has 34 attributes described in figure 2. It has several raw network packets in it. The dataset contains six major attack categories such as Bruteforce, DoS, Flood, legitimate, malformed and slowite. This dataset is divided into a test set with 30% of the records and a training set with 70% of the records.

0	tcp.flags	330926 non-null	object
1	tcp.time_delta	330926 non-null	float64
2	tcp.len	330926 non-null	int64
3	mqtt.conack.flags	330926 non-null	object
4	mqtt.conack.flags.reserved	330926 non-null	float64
5	mqtt.conack.flags.sp	330926 non-null	float64
6	mqtt.conack.val	330926 non-null	float64
7	mqtt.conflag.cleansess	330926 non-null	float64
8	mqtt.conflag.passwd	330926 non-null	float64
9	mqtt.conflag.qos	330926 non-null	float64
10	mqtt.conflag.reserved	330926 non-null	float64
11	mqtt.conflag.retain	330926 non-null	float64
12	mqtt.conflag.uname	330926 non-null	float64
13	mqtt.conflag.willflag	330926 non-null	float64
14	mqtt.conflags	330926 non-null	object
15	mqtt.dupflag	330926 non-null	float64
16	mqtt.hdrflags	330926 non-null	object
17	mqtt.kalive	330926 non-null	float64
18	mqtt.len	330926 non-null	float64
19	mqtt.msg	330926 non-null	object
20	mqtt.msgid	330926 non-null	float64
21	mqtt.msgtype	330926 non-null	float64
22	mqtt.proto_len	330926 non-null	float64
23	mqtt.protoname	330926 non-null	object
24	mqtt.qos	330926 non-null	float64
25	mqtt.retain	330926 non-null	float64
26	mqtt.sub.qos	330926 non-null	float64
27	mqtt.suback.qos	330926 non-null	float64
28	mqtt.ver	330926 non-null	float64
29	mqtt.willmsg	330926 non-null	float64
30	mqtt.willmsg_len	330926 non-null	float64
31	mqtt.willtopic	330926 non-null	float64
32	mqtt.willtopic_len	330926 non-null	float64
33	target	330926 non-null	object

Figure 2 Attributes of MQTT dataset for SIoT

Dataset Preprocessing

Preprocessing the data involves identifying any missing values within the dataset, which can be achieved using the null().sum() function in the Pandas library. This function helps count the occurrences of missing values. This data set appears to be missing no values upon review. After missing imputation, the data is pre-processed using Robustscaler and label encoder to handle scaling the all-variable unit as unique. Scikit-Learn offers the Label Encoder class for this reason, which allows you to convert all string values to float values. It provides a unique numerical value for every category in a variable, making it easier for ML algorithms to examine and comprehend the data. After removing the median, RobustScaler scales the data using the quantile range. The dataset contains a total of 4115 with 14 attributes. Figure 3 displays the pre-processed data.

#	Column	Non-Null Count	Dtype
0	tcp.flags	4115 non-null	int32
1	tcp.time_delta	4115 non-null	float64
2	tcp.len	4115 non-null	int64
3	mqtt.conack.flags	4115 non-null	int32
4	mqtt.conack.val	4115 non-null	float64
5	mqtt.conflag.uname	4115 non-null	float64
6	mqtt.dupflag	4115 non-null	float64
7	mqtt.kalive	4115 non-null	float64
8	mqtt.len	4115 non-null	float64
9	mqtt.msgid	4115 non-null	float64
10	mqtt.msgtype	4115 non-null	float64
11	mqtt.proto_len	4115 non-null	float64
12	mqtt.retain	4115 non-null	float64
13	target	4115 non-null	int32

dtypes: float64(10), int32(3), int64(1)
memory usage: 402.0 KB

Figure 3 Pre-processed attributes for SIoT

The heat distribution map, which is displayed in Figure 4 displays the correlation matrix between the 21 elements in the MQTT dataset based on IoT sensors. Figure 5 displays correlation analysis between 14 factors MQTT dataset with dropping out few datasets.

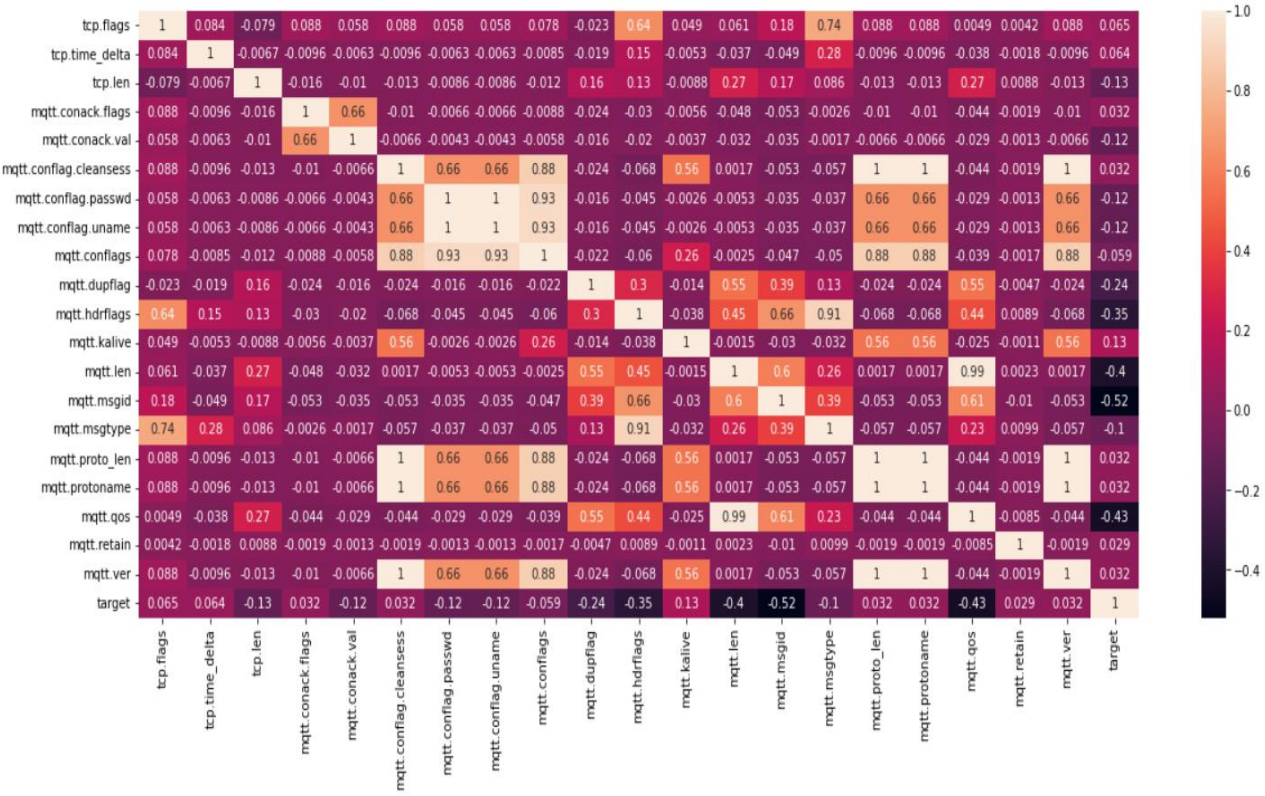

Figure 4 Correlation analysis between 21 factors MQTT dataset based on IoT sensors

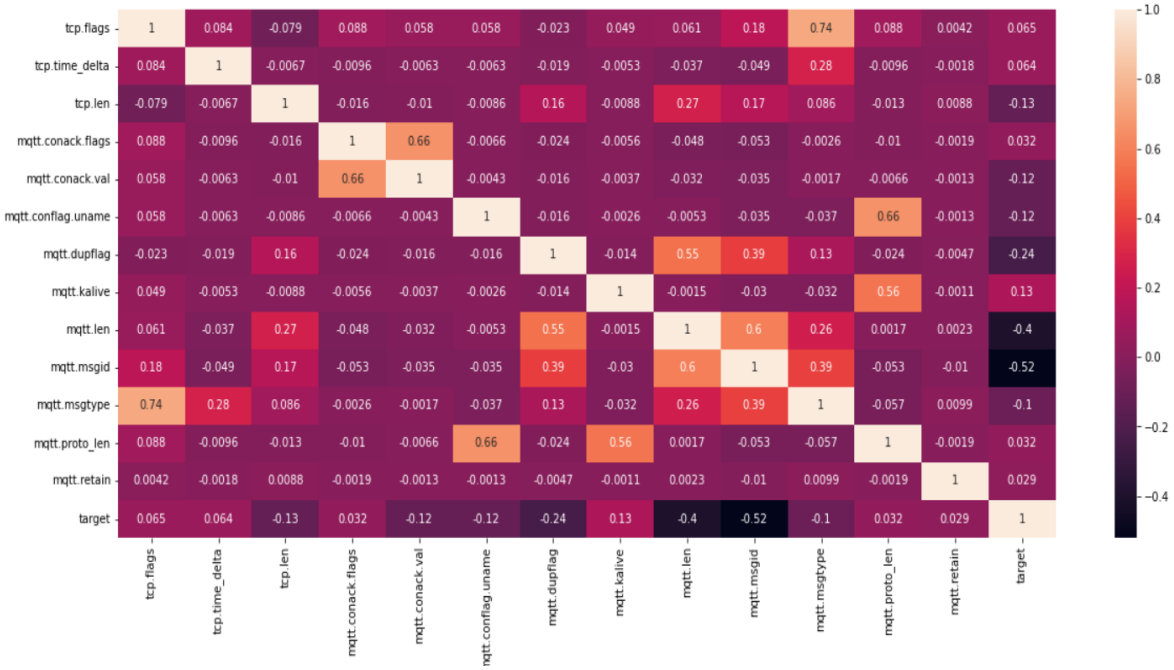

Figure 5 Correlation analysis between with dropping out few datasets

Research Methodology

In order to detect malware attacks early and stop system tampering, a strong classification model that can distinguish between malicious and benign traffic in SIoT networks is needed. This research work proposed OPTUNA based TPE framework has cogitates the best methodical approach due to it tuning parameters like learning rate, loss functions, etc. Optuna is user-friendly, well-built software that works with a range of optimization's techniques. The Python language is used to develop and train this mode with the help of several libraries, including Matplotlib, NumPy, Scikit-Learn, and Pandas. After being cleaned, the real-time database's data is exported as a CSV file. The functions needed to carry out the hyperparameter optimizations are provided using Scikit-Learn module. The overall proposed model is as shown in figure 6.

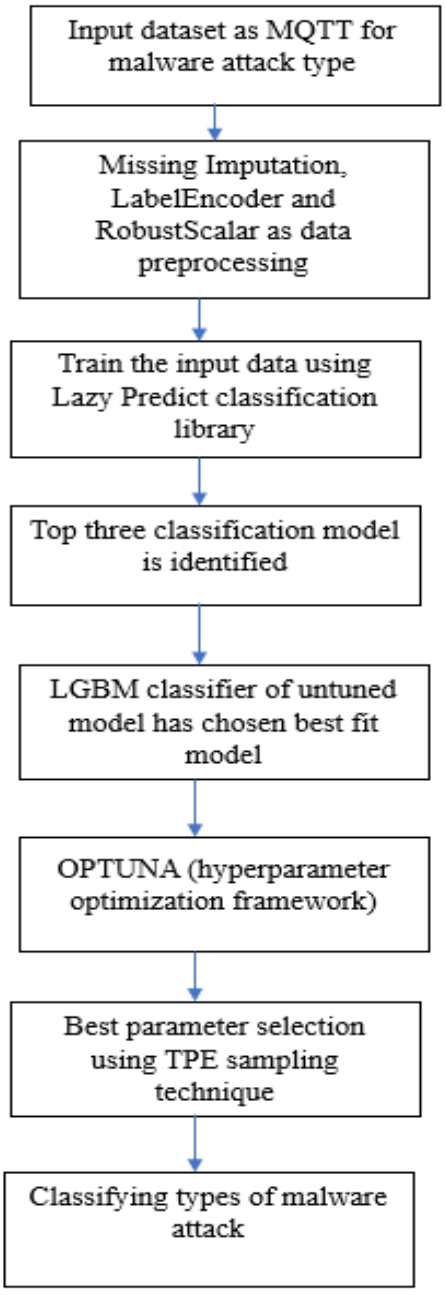

Figure.6 Proposed architecture for malware attack detection in SIoT

This study employs an ML classifier model that is created by grouping different classifiers into a single supervised Lazy Predict library. The dataset is split in term of models and predictions with respect to train dataset and test dataset. The classification models are made to be evaluated using Lazy Predict supervised Lazy Classifier library. Lazy Classifier provides a convenient and efficient way to fit and evaluate multiple ML models, simplifying the model selection process and allowing to focus on building the best model for our data. Lazy Predict aids in the development of multiple, distinct, fundamental ML models using specific code and helps to identify which models may perform more accurately avoiding the need for parameter tuning. In this study, regression-based datasets are solved using a Lazy Classifier in classifying the problem with better identification of malware attack in SIoT from MQTT dataset. In order to employing the best readily accessible ML model, the Lazy Predict classifier can be optimized for accuracy. This can be achieved by optimizing the top model's hyperparameters, as suggested by the OPTUNA hyperparameter tuning. This research work LGBM classifier, ExtraTree Classifier (ETC) and XGB Classifier has chosen the top three Lazy Predict model to determining the attack type. Further, this LGBM classifier of untuned model has chosen best fit models based on highest accuracy value which can be improved using fine tuning of OPTUNA based TPE sampler framework.

According to OPTUNA, optimising hyperparameters involves minimising or maximising an objective function that receives a collection of hyperparameters as input and outputs the function's (validation) score. It is this function that dynamically generates the neural network architecture's search space (the number of layers and hidden units) without depending on externally supplied static constants. A trial object is a special OPTUNA trace object that looks for the ideal value based on the hyperparameter's name and range. Through interaction with a trial object, OPTUNA gradually constructs the target function and, when the target function is being executed, dynamically produces the search space using the trial object. There are two types of sampling methods in OPTUNA: independent sampling and relational sampling. The correlations between parameters are exploited by relational sampling. The relationships between the parameters were not taken into consideration by independent sampling. Depending on the work and surroundings, both relational and independent sampling can be cost-effective. When combined with these two sampling algorithms, OPTUNA can handle independent sampling techniques like TPE, resulting in faster and more effective convergence. Using past hyperparameter evaluations, TPE is an iterative procedure that generates a probabilistic model. The next set of hyperparameters to be assessed is then recommended by the model. After that, switch to a different modeling technique and EI optimisation strategy for the SMBO algorithm, assuming that our hyper-parameter optimisation tasks will require large dimensions and small budgets for fitness evaluation. When dealing with hyperparameter optimization where the budget for fitness evaluation is constrained and there are larger dimensions, an alternative to the Gaussian process (GP) approach is required. Estimating f(x), or the likelihood of a score given the hyperparameter, was the aim of GP. P(b|a) using marginals that took into account the probability of each hyperparameter given the score and the likelihood of each hyperparameter:

$$P(b|a) = \frac{P(b)\ X\ P(b)}{P(a)} \qquad \text{.....(1)}$$

where a stands for the hyperparameters and b represents the score f(a). But in TPE, instead of approximating the left side of equation (1), An approximation of P(a|b), the probability of the hyperparameters assuming the score received when sampling certain of the hyperparameter values, is attempted. Two distinct functions are used to approximate this conditional probability (P(a|b)): function i (a) for situations where performance is less than that value, function j(b) for situations when performance exceeds a predetermined value of performance:

Two such densities are used by the TPE to define p(a|b) in equation 2.

$$p(a|b) = \{i\ (a),\ if\ b < b*\ j(b),\ if\ b\ \geq b* \qquad \text{.....(2)}$$

these two densities of l(a) and g(b) will then be used in the EI function

i (a) is the density that results from utilising the data. { a^k } such that corresponding loss

f { a^k }was less than $b*$ and g(a) is the density that results from utilising the leftover observations. By altering the generating process and substituting non-parametric densities for the configuration prior's distributions, the TPE models p(a|b).

Algorithm:

Input: Training dataset MQTT for SIoT, Testing dataset MQTT for SIoT
Output: Classification results
Step 1: Establish a function that needs to be minimised.
Step 2: To create hyperparameters, use a Trial object's suggested methods.
Step 3: a benchmark associated with the Trial object.
Step 4: Bring up the goal function's optimisation.
Step 5: Construct an objective function that accepts hyperparameters and returns a score that we wish to minimise, such as loss, root mean squared error, or cross-entropy.
Step 6: A few observations (a score) are obtained using a randomly chosen set of hyperparameters.
Step 7: After classifying the gathered observations according to a quantile, split them into two groups. The observations that received the highest ratings are in the first group (a), and all other observations are in the second group (b),
Step 8: Parzen Estimators, sometimes referred to as kernel density estimators, are a straightforward means of averaging kernels centred on available data points to represent two densities, i(a) and j(b).
Step 9: Suggest hyperparameters from i(a), assess them using i(a)/j(b), and get the set that produces the lowest value under i(a)/j(b) that most closely aligns with the anticipated improvement. The objective function is then used to evaluate these hyperparameters.
Step 10: Revise the step 6 observation list.
Step 11: Steps 7–10 are repeated until the time limit is met, or a set number of tries are made.

The proposed work describes untuned lazy predict LGBM model were optimized and the loss function is improved. The OPTUNA based TPE framework was utilized to improve the hyperparameters, and the primary loss function was utilized as the enhanced loss function. This research work, hyperparameter tuning of certain parameters are used to get the best hyperparameter. The seven hyperparameters, including lambda_l1, lambda_l2, num_leaves, feature_fraction, bagging_fraction, bagging_freq, min_child_samples were chosen for parameter optimization to get best value. Table 1 displays the seven hyperparameters for LGBM model were chosen as a best trial value optimization using an OPTUNA-based TPE sampler.

Table.1 Optimized hyperparameter for LGBM model using OPTUNA-based TPE sampler.

S.No	hyperparameter name	Tuned parameter
1	lambda_l1	1.40386
2	lambda_l2	0.56347
3	num_leaves	34
4	feature_fraction	0.693966
5	bagging_fraction	0.77316
6	bagging_freq	5
7	min_child_samples	23

The parameters of LGBM are described as follows:

Lambda_l1 and Lambda_l2 specifies L1 or L2 regularization. Since the magnitude of these factors is not directly correlated with overfitting, it is more difficult to determine the ideal value. For both, despite this, a decent search range is (0, 100).

bagging_fraction determines the percentage of training samples that will be used to train each tree by taking a value between 0 and 1.

feature_fraction specifies the proportion of characteristics to sample for each tree's training. It so requires a value in the range of 0 and 1.

num_leaves, the most crucial factor governing the tree structure. As the name implies, it regulates the quantity of decision leaves present in a single tree. The node where the "actual decision" is made in a tree is called the decision leaf. the maximum limit to num_leaves should be 2^(max_depth)

bagging_freq is the rate of data sampling frequency. Other tree-based models default to resampling the data before each tree when this parameter is set to 1. The subsample's range is from 0.05 to 1.

min_child_samples Minimum number of data needed in a child (leaf).

Results and Discussion

In this study, Jupiter IDE and Google Colab are used to produce and distribute documents that can be explained using text, live code, and visualizations. In addition, the tunability hyperparameter made use of Seaborn, Sklearn and Pandas. The SIoT based MQTT dataset and it can be evaluated through confusion matrix metrics measure and compared with various classifier using a single library named lazy predict classifier. This library assists in training the data pre-processed sample that has been split as 70% as train dataset and 30% as test dataset. The description of malware attack type with corresponding python code representation is as: 0 - 'Bruteforce', 1 – 'DoS', 2 – 'Flood', 3 – 'Legitimate', 4 – 'Malformed', 5 – 'slowite'. The different performance metrics are precision, recall, f1 score, and support. Figure.7,8,9 and 10 illustrates the confusion matrix class value for top three lazy predict classifier and tuned LGBM classifier based on attack labels.

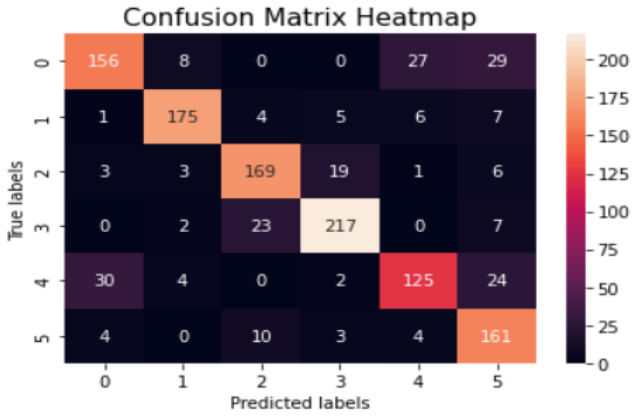

Figure.7 Confusion matrix for LGBM classifier

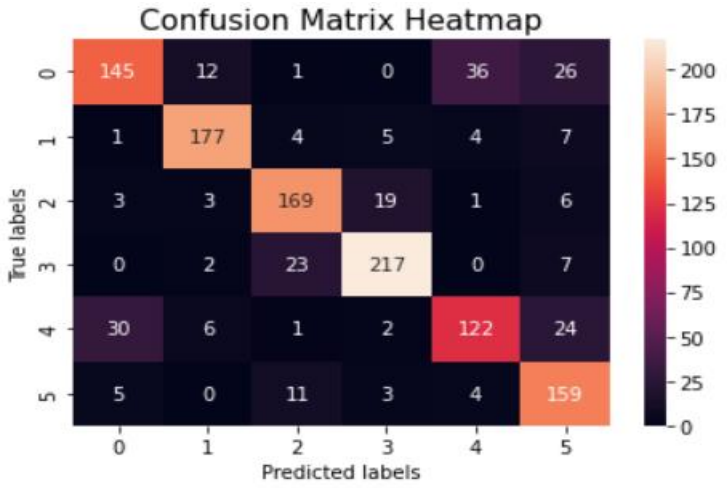

Figure.8 Confusion matrix for ExtraTree classifier

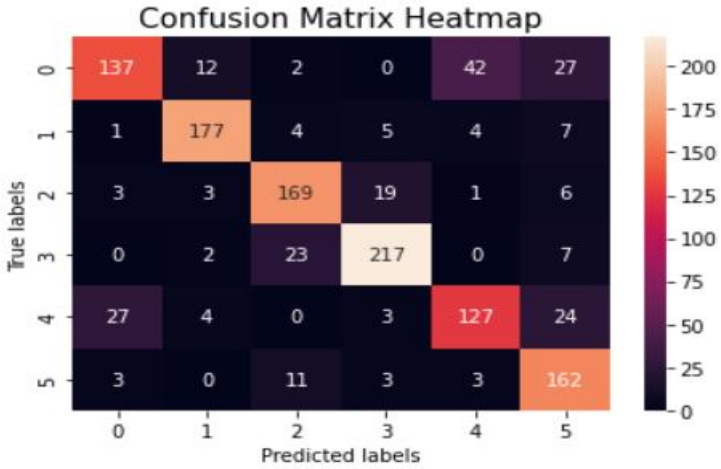

Figure.9 Confusion matrix for XGB classifier

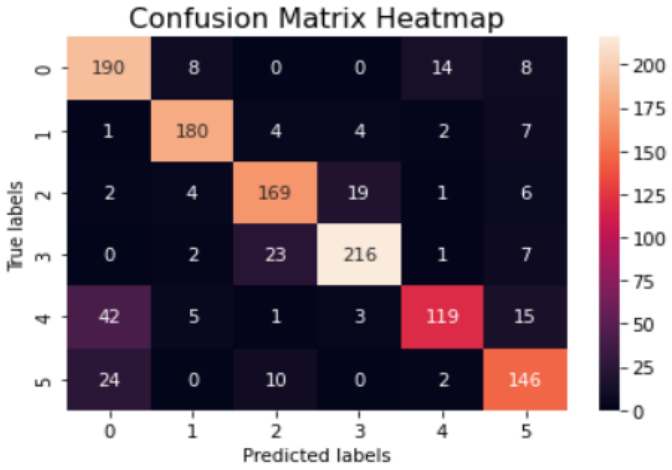

Figure.10 Confusion matrix for proposed tuned LGBM classifier

Figure 11 illustrates the micro and weighted metrics in which accuracy can be determined through micro precision, micro recall and micro f1-score. The value of micro f1-score is said to be accuracy in proposed tuned LGBM classifier has high accuracy as 0.83 while compared with top three lazy predict model of LGBM, ExtraTree and XGB classifier as 0.81, 0.80 and 0.80 correspondingly. Similarly, the individual label weight is measured and estimated through weighed precision, weighted recall and weighted F1-score in figure 12.

	precision	recall	f1-score	support
0	0.73	0.86	0.79	220
1	0.90	0.91	0.91	198
2	0.82	0.84	0.83	201
3	0.89	0.87	0.88	249
4	0.86	0.64	0.73	185
5	0.77	0.80	0.79	182
accuracy			0.83	1235
macro avg	0.83	0.82	0.82	1235
weighted avg	0.83	0.83	0.82	1235

(0.8259109311740891, 0.8210723695978146)

Figure.11 Weighted average calculation for all the attack type

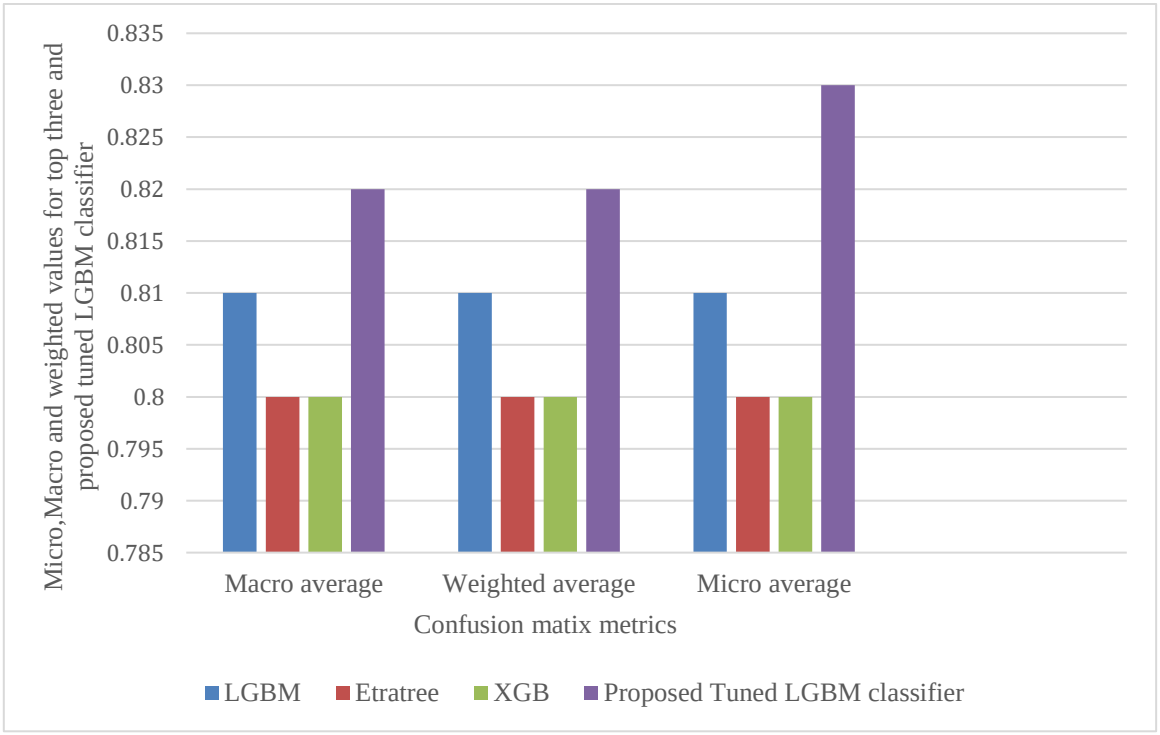

Figure.12 Micro, Macro and Weighted values for various classifiers

According to the experimental results in figure 12 it determined weighted precision, weighted recall and weighted F1-score is 0.83 respectively which is better in determining the attack type accurately than top three lazy predict classifier.

Conclusion

IoT malware detection is becoming more and more crucial to safeguarding users' personal data and the SIoT infrastructure. Appropriate and reliable procedures that work in a variety of situations are necessary for effective malware identification. Network security depends on the deployment of autonomous systems, however approaches such as centralized or distributed detection have trade-offs with performance. This research work uses OPTUNA based TPE to identify attacks like DOS, flood, slowite, brute force and malformed attacks. The parameter analysed here is precision, recall, f1 score, and support. Our study showcased a proposed tuned LGBM classifier achieving 0.83 accuracy in SIoT malware detection than the top three lazy predict classifier. As a result, while our research provides a strong approach for SIoT malware detection, more research is necessary to strengthen network security against new threats. Additionally, improving distributed detection strategies and including predictive analysis are also essential. The foundation for additional research paths aimed at developing more adaptable and resilient security mechanisms for SIoT ecosystems is laid by this work.

Reference

[1] IDC, “The Growth in Connected IoT Devices,” 2019. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS45213219

[2] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, “A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues,” IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1457–1477, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7900337/

[3] H. Ning, F. Farha, Z. N. Mohammad, and M. Daneshmand, “A Survey and Tutorial on “Connection Exploding Meets Efficient Communication” in the Internet of Things,” IEEE Internet of Things Journal, vol. 7, no. 11, pp. 10 733–10 744, nov 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9098906/

[4] H. Ning, F. Shi, S. Cui, and M. Daneshmand, “From iot to future cyberenabled internet of x (iox) and its fundamental issues,” IEEE Internet of Things Journal, 2020.

[5] Q. Du, H. Song, and X. Zhu, “Social-feature enabled communications among devices toward the smart iot community,” IEEE Communications Magazine, vol. 57, no. 1, pp. 130–137, 2018

[6] M. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, “Social internet of things (siot): Foundations, thrust areas, systematic review and future directions,” Computer Communications, vol. 139, pp. 32–57, 2019.

[7] Claudio Marche, Luigi Atzori, Virginia Pilloniis, How to exploit the social internet of things: query generation model and device profiles dataset, Comput. Netw. (2020) 107248.

[8] Mishra, D., et al., SEM: Stacking ensemble meta-learning for IOT security framework. 2021. 46(4): p. 3531-3548.

[9] Nookala Venu, Dr, AArun Kumar, and Mr A. Sanyasi Rao., ”Botnet Attacks Detection In Internet Of Things Using Machine Learning.”, Neuroquantology, 20.4, 743-754 (2022)

[10] Nguyen, Giang L., et al., ”A collaborative approach to early detection of IoT Botnet.”, Computers & Electrical Engineering, 97, 107525 (2022)

[11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, ‘‘OPTUNA: A nextgeneration hyperparameter optimization framework,’’ in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Anchorage, AK, USA, 2019, pp. 2623–2631.

[12] Piragash Maran, Timothy Tzen Vun Yap, Ji Jian Chin, Hu Ng, Vik Tor Goh, and Thiam Yong Kuek, “Comparison of Machine Learning Models for IoT Malware Classification”, CITIC 2022, 10, pp. 15–28, 2022.

[13] Ali Mehrban and Pegah Ahadian, “Malware Detection in IoT Systems Using Machine Learning Techniques”, International Journal of Wireless & Mobile Networks (IJWMN), Vol.15, No.6, December 2023.

[14] Hyunjong Lee, Sooin Kim, Dongheon Baek, Donghoon Kim, And Doosung Hwang, “Robust IoT Malware Detection and Classification Using Opcode Category Features on Machine Learning”, IEEE Access, Volume 11, 2023.

[15] J. Jeon, J. H. Park, and Y.-S. Jeong, ‘‘Dynamic analysis for IoT malware detection with convolution neural network model,’’ IEEE Access, vol. 8, pp. 96899–96911, 2020.

[16] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, ‘‘Federated learning for malware detection in IoT devices,’’ Comput. Netw., vol. 204, Feb. 2022, Art. no. 108693.

[17] M. Dib, S. Torabi, E. Bou-Harb, N. Bouguila, and C. Assi, ‘‘EVOLIoT: A self-supervised contrastive learning framework for detecting and characterizing evolving IoT malware variants,’’ in Proc. ACM Asia Conf. Comput. Commun. Secur., May 2022, pp. 452–466.

[18] Sahraoui Dhelim, Huansheng Ning, Fadi Farha, Liming Chen, Luigi Atzori and Mahmoud Daneshmand, “IoT-Enabled Social Relationships Meet Artificial Social Intelligence”, IEEE IoT JOURNAL, 2021.

[19] A. Khelloufi, H. Ning, S. Dhelim, T. Qiu, J. Ma, R. Huang, and L. Atzori, “A Social Relationships Based Service Recommendation System for SIoT Devices,” IEEE Internet of Things Journal, pp. 1–1, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/ 9167284/

[20] Abdelghani W, Zayani C.A, Amous I, Florence S. Trust Management in Social Internet of Things: A Survey. Dwivedi Y. et al. (eds) Social Media: The Good, the Bad, and the Ugly. I3E 2016. Lecture Notes in Computer Science, 9844, pp 430–441, 2016

[21] Roopa MS, Pattar S, Buyya R et al (2019) Social Internet of Things (SIoT): foundations, thrust areas, systematic review and future directions. Comput Commun 139:32–57.

[22] Ahmed M. Elshewey, “hyOPTGB: An Efficient OPTUNA Hyperparameter Optimization Framework for Hepatitis C Virus (HCV) Disease Prediction in Egypt”, Research Square, 2023, https://doi.org/10.21203/rs.3.rs-2768795/v1.

[23] Shuhei Watanabe, “Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance”, arXiv:2304.11127v3 [cs.LG] 26 May 2023.

[24] Shuhei Watanabe and Frank Hutter, “c-TPE: Generalizing Tree-structured Parzen Estimator with Inequality Constraints for Continuous and Categorical Hyperparameter Optimization”, NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems,2022.