Atomic Spectroscopy

Article

Malware detection in Social Internet of Things (SIoT) for E-commerce using Machine Learning methods

Anciline Jenifer J, Piramu Preethika S.K
Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS)
Department of Computer Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS)

ABSTRACT: The Internet of Things (IoT) plays a key role in each object of the next-generation people like smartphones, wearable devices as well as actuators and sensors have been provided with digital counterparts. The goal of augment the ability of physical objects and perform on behalf of communicating with third parties. The object has ability in interacting and establishing autonomous social relationships in accordance with the Social Internet of Things (SIoT). Objects such as humans have been considered to be social and intelligent. They created its Social Network (SN) to accomplish their usual goals like improvement in performance, functionality and efficiency for satisfying their needed services. Their privacy might be violated and their data can be made available to the public. IoT is unlikely to take the lead as a technology until it has proven methods to strengthen reliable connectivity between nodes. There are various preventions of malware detection have been created subsequently to hide their hazardous behaviors from analysis tools. Therefore, it's unable to use traditional malware detection techniques, and the SIoT must be secured by creative solutions against such anti-detection malware. In identifying malware attacks imposed by hostile nodes as well as separating it from the network. This proposed Autoencoder (AE) utilizes the auxiliary data for pre-trained and Machine Learning (ML) is using for fine tuning model for accurate detection of malware from the Message Queuing Telemetry Transport (MQTT) dataset. To evaluate the proposed AE with ML method has accomplished the best performance using confusion matrix metrics such as accuracy, precision, recall, F1-score are comparing with an existing methodologies.

Keywords: Internet of Things, Social Internet of Things (SIoT), Autoencoder, malware detection, security, Machine Learning (ML).

INTRODUCTION

The IoT idea has currently come into existence as globalization as well as smart device connectivity with skyrocketed. Despite involving humans, IoT devices are utilized in sensing, controlling, monitoring intelligent data sharing, and other functions. The idea of the IoTs has been used in smart grids for transportation and agriculture, smart cities, smart public safety systems and smart healthcare among other smart global networks. Featuring billions of gadgets capable of sending important data about the physical environment and carrying out basic tasks of IoT has begun to pass and provide rise to the vision of anytime and anywhere connectivity with anything [1]. There are several areas like smart building, supply chain of logistics and transportation, industrial production plant management and e-health have developed recent applications with large amounts of data flow using IoT. In addition, there are some IoT applications that are created for an architecture of service-oriented in which each device is performed as a service requester or service provider else both activity. IoT has forwarded towards the model of things that focused on other things in providing combined services to the human being benefits. Understanding how any object's information can be effectively evaluated by any additional peer in the system is crucial when using a similar relationship model. The requester, who plays the role of the trustor in the IoT scenario, must have faith that the supplier, who then plays the role of the trustee, will deliver the requested service. However, malfunctioning devices are performed with different assaults while compared to benefits of other IoT nodes. Hence, fake suggestion or services have been provided to perform solely or organizing cooperating community devices for controlling hre service classes. Thus, the attacks as well as malfunctioning could completely negate any IoT advantages if they aren't get handled effectively [2] [3].

Consequently, mobile, web, and sensor technologies have quickly developed and are being integrated. This technology may now provide physical items including doors, home applicants that turn into "smart things" over digital era. Gartner and Cisco Systems predict that there are going to be 20 billion and 50 billion linked devices, correspondingly by the year 2020. This might raise the degree of IoT network complexity in the future. The present IoT integration techniques don't adhere to common design standards aren't economically viable, and hinder the full potential of IoT from being realized. The concept of SN and the IoT share certain commonalities [4]. The SN research is able to be utilized for tackling implementation issues in the IoT environment. As a result, social ties between items in IoT networks serve as the foundation of the new paradigm known as SIoT. Smart objects are connected using devices with occasional connections dependent on user relations using relationships with others including parental, shared facilities, co-work, ownership, and social objects. SIoT aims in maintaining the two levels of individuals as well as things separated permitting the objects to establish independent SN. This allows individuals to establish rules for safeguarding its privacy while limiting access to the autonomous results of inter-object interaction operating on the SN objects [5]. The SN for smart devices also makes it possible to do the following

functions in a secure setting, network navigability, interoperability and service discovery. In addition, Ericsson researchers have suggested utilizing SN as an intellectual framework or analogy to describe the intricacy of user-thing interactions and IoT integration. It provides to SIoT which began about 2011 and is currently employed interchangeably. Despite the fact of SIoT is still early stages, extensive research has currently done to provide concepts for integration in practical applications [6]. Even though the concept of social relationships between objects initially emerged as SIoT it applied in several ways depending on the study. There are hardly any investigations that have examined various ideas of SN in SIoT and consider the growing interest in SIoT as a research issue.

The IoTs are envisioned as SN with features for collaboration as well as community engagement. Each object can connect data and exchange with other devices as well as computing systems using the Internet as well as additional networks using sensors, software, processors, and other technologies that have been integrated. These social connections enable interaction between devices and individuals to enhance their SN and make sharing data easier. IoT reuses the principles of human social networking to address the issues of IoT. The models currently employed to analyze human SNs have been used for resolving IoT-related problems [7]. Although security solutions have evolved the issues with scalability, centralization, as well as an unclear designing. Additionally, they overemphasize the physical features of the gadgets while ignoring the output obtained from relationship among gadgets and consumers as social intelligence. Thus, the trust has built through such environment connections such as families, enterprises, and friendships. These solutions fail to improve actual interactions between people. Devices connected to the IoTs are particularly vulnerable to network attacks such as fraud, DDoS attacks and spoofing. The methods of attack and vulnerabilities like new attacks, botnets, and additional forms of cyberattacks are utilized.

The SIoT paradigm makes use of arbitrary behavioral characteristics to guarantee accurate data analysis, expert services, and enhanced security [8]. There are certain research gaps that have been fulfilled through efficient scalability, object discovery and efficiency which is similar to the platform of human social networking management between network navigability and smart social devices in the phenomenom of smart world context by management intraction between reliability and objects for the user's smart devices [9]. The existing IoT technologies has combined with human social activities named SIoT system that has an opportunity to offer users ubiquitous connectivity. The objectives of SIoT systems change from distributing data to user delight, computation offloaded is crucial to speed up the execution of program [10]. Delivering excellent service in a fully protected environment becomes significantly more difficult, adding to the existing challenging responsibility of securing sensitive data. A few studies have made an effort to investigate this issue. They put forth a wide range of models that categorize trusted nodes in the IoT network using various criteria and aggregation techniques. However, the earlier efforts have failed to offer any strategies for identifying fake nodes or discriminating between assaults. Hence, the proposed AE based ML is used to improve accuracy of detecting malware attacks efficiently.

The structure of this paper is explained as follow, session 2 discusses about detecting malware attack using ML and AE methods. Session 3 illustrates the proposed AE with LGBM classifier for pre-training the MQTT input and fine tuning LGBM classifier for better classification. Session 4 illustrates the performance evaluation of AE with LGBM classifier using confusion matrix metrics with existing methods. Session 5 concluded that AE with LGBM classifier has high accuracy in detecting malware attack effectively.

LITERATURE REVIEW

In SIoTs model, each node is considered as an entity that interact socially with other objects individually in accordance with established rules by their owner. This idea is quickly gaining traction due to the major advantages resulting from the SN possibilities in the IoT domain like simplifying the dynamic network navigations. The billion object efficiency in the dynamic discovery, selection, as well as services composition offered through dispersed networks and objectss, reliable management of the object's constancy while maintaining services and data.

R. Chen et al. have suggested an adaptable mechanism of decentralized trust with respect to social trust is one of the efforts taking social considerations into account. It combines collaboration and social community variables through a weighted and it utilizes two actual social IoT scenarios to illustrate the model's efficacy [11]. K. Zhao and L. Pan have presented a further trust model pertaining to social trust that suggests formalizing the trust assessment as a classification issue using a ML-based technique. In a social network, social parameters including reputation as well as centrality are used for developing the feature vector [12]. A. M. Kowshalya and M. Valarmathi have presented a trust management strategy using certain metrics of SIoT trust, such as centrality, community interest, and cooperation, to promote autonomous trustworthy decision-making that relies on smart device behavior [13]. The value of the service as well as metrics like social similarity have been taken into consideration by B. Jafarian et al. The ensuing trust management algorithm calculates the nodes trust level in a SIoT network by social relationships. A centralized trust-based system for moving items has been presented by R. Chen et al. and relationships with others are used by the system to ensure accuracy and trust among the devices. [14] [15].

To elaborate on the advantages of SIoT over traditional IoT and the research team highlighted the benefits of networking "social objects" as opposed to "smart objects," that regarded as a generational shift from objects in few degree of intelligence for the objects with actual consciousness in social [16]. Khelloufi et al. have proposed a recommendation system for services that obtain advantage of the social connections among individuals of IoT devices. The recommendation depends on the various connections among service provider and service requester. It also involve detection algorithm in boundary-based community which utilized for creating communities of social connected device owners [17]. Deep learning models and ML-based malware detection techniques are the key methodologies used in the model training as well as classification phase. Wang et al. have used five ML models to perform software classification, including Support Vector Machine (SVM), Naive

Bayes (NB), K-Nearest Neighbour (KNN), Random Forest (RF), and Classification Regression Tree (CART), the methods that utilize algorithms primarily employ typical ML and classification models [18]. A feature learning model incorporating a variety of ML methods was suggested by Kumar et al. [19] to identify malware with high accuracy and minimum overhead. W. Wang et al. [20] created a hybrid approach based on deep AE as a pre-trained technique and several CNN structures in malware identification in an effort for improving the precision of Android malware detection with large-scale. The CNN-P structure achieved the best accuracy based on experimental results. Yi Zhang et al. [21] have developed a system called DeepClassifyDroid and presented an identification strategy based on Convolutional Neural Networks (CNN). Three parts make up DeepClassifyDroid's structural layout: the feature extraction component, the embedding in vector space and the DL model which uses CNN for malware characterization.

RESEARCH METHODOLOGY

The concepts of social networking and IoTs were combined to create the SIoT paradigm. It permits communication between linked equipment and people and offers a wide range of intriguing applications. The introduction of IoT into telecommunications environments initiated an evolutionary process that resulted in SIoT. The dataset is composed by 8 MQTT sensors with different features. In table, the MQTT sensors are reported. Each sensor has a data profile, as well as a topic connected to the MQTT broker. The subject is specified by the sensor when transmitting the data to the broker, and the data profile describes the kind of data the sensors transmit. This concept is important since a temperature sensor has a periodic behavior over time, i.e. cyclically sending information retrieved from the environment periodically (defined as P). Instead, a motion sensor has a more random behavior since it sends information only when a user passes in front of the sensor (defined as R)). By analyzing also this aspect, the dataset is even more valid as a real behavior of a home automation is simulated and implemented.

Dataset collection

The proposed effort seeks to produce an initial dataset for the scientific and industrial communities to utilize their applications that are related to the IoT context, focusing specifically on the MQTT communication protocol. Each component of a real network is defined in the dataset, which is made up of IoT sensors based on MQTT. In precisely, the network consists of 8 sensors and the MQTT broker is created using Eclipse Mosquitto. In the scenario, 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.

tcp.flags	tcp.time_	tcp.len	mqtt.cona	mqtt.cona	mqtt.cona	mqtt.cona	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.conf	mqtt.dup	mqtt.hdrf	mqtt.kaliv	mqtt.len	n
0x0000001	1.90E-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	0	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	165	
0x0000001	1.00E-06	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000004	0	2	
0x0000001	1.00E-06	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	165	
0x0000001	4.00E-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	3.00E-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	0.000448	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	6.10E-05	1460	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	172	
0x0000001	9.30E-05	1460	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	166	
0x0000001	0.00012	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	12	
0x0000001	0.000337	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	9.10E-05	492	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	163	
0x0000001	4.00E-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	5.00E-06	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	11	
0x0000001	0.001751	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000004	0	2	
0x0000001	0.000133	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	10	
0x0000001	60.0001	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0x000000c	0	0	
0x0000001	2.00E-06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0000001	1.00E-06	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000004	0	2	
0x0000001	4.10E-05	684	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	169	
0x0000001	0.000148	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	8	
0x0000001	8.90E-05	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0000003	0	11	
0x0000001	3.10E-05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 1 MQTT dataset as an input for SIoT

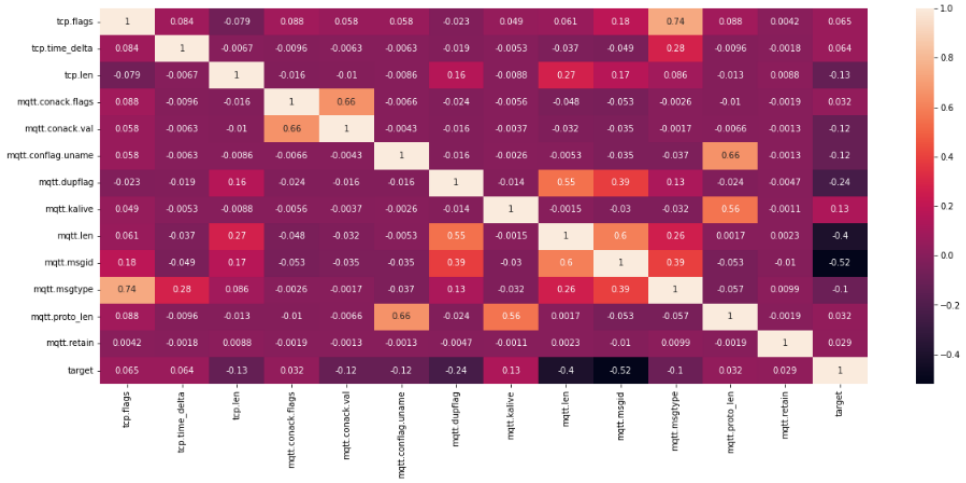

Figure 2 Correlation heatmap for various features in MQTT dataset

Python Code Representation for attack type	Attack type description as target
0	Bruteforce
1	Dos
2	Flood
3	legitimate
4	malformed
5	slowite

Table 1 Code representation for various attack type as target

WORKING OF STACKED AUTOENCODER

One of the unsupervised Neural Network (NN) is AE that assist to learn for reducing the difference among input data as well as output data. This AE consists of two type namely encoder and decoder. The original data gets mapped with encoder code which basically deals with code dimension is lesser than an original data. In the case of decoder, the code has tried to map an original input. The dimensionality reduction is an AE application and consider input as $z \in R^n$, the AE goal is represented as $y = z$ which tried for learning AE function is expressed in equation 1.

$$F_{w,ib}(x) \approx y \quad (1)$$

Where,

W = Weight of the entire neural network

ib = Image bias

The basic reconstruction loss in AE loss function for L_p distance in which Stochastic Gradient Descent (SGD) has been utilized to fine tune the weight and bias in AE module shown in equation 2.

$$\mathbb{L}(w, ib) = min\left\|x - F_{w,ib}(x)\right\|_p \quad (2)$$

However, the better results are obtained through AE that involves various AE in which the output of each AE is assigned to the input of the succeeded AE. The given below steps are basic steps for AE training.

Step 1 - Encoder transformation

The AE with M number is represented as m^{th} AE's encoder as well as decoder transformation functions. The function of encoder transformation in AE has evaluated using function of encoder transformation for each AE in forward order gets illustrated in equation 3.

$$x_{encoded} = x^m = \alpha^m, \alpha^{m-1}, \dots, \alpha^2, \alpha^1\ (x) \quad (3)$$

Step 2 - Decoder transformation

In the case of AE decoder transformation function has been evaluated by function of decoder transformation for each AE in reverse order get illustrated in equation 4.

$$x_{Decoded} = x_{reconstrct} = \alpha^1, \alpha^2, \dots, \alpha^2, \alpha^1(x^w) \quad (4)$$

When one layer is trained, the other layer's parameters get fixed whereas the output of the preceding layer has been utilized as an input for the subsequent layer. Thus, it will continue till the training gets completed. The backpropagation algorithm has been utilized for reducing the reconstruction error once all the layers are trained and all the layer's weights get modified.

Working of Light Gradient Boosting Machine (LGBM) classifier

One of the gradient boosting framework is LGBM that completely relies on Decision Tree (DT) for improving the model efficiency and minimize the usage of memory. This method includes two novel techniques namely Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) in which mean of GOSS and EFB with Gradient Boosted Decision Tree (GBDT) in which GDBT is expressed in equation 5.

$$F(x, w) = \sum_{t=0}^{T} \alpha_t h_t(x, w) \quad (5)$$

Where,

F( ) = Predictive value for GBDT

$h_t(\ )$ = t-th DT method function
w = DT parameters
x = Input samples
α = weight of each tree

When the loss function minimization L( ) to map the space x and space y. The optimal model get solved is expressed in equation 6.

$$\hat{F} = \arg\min_{F} \sum_{i=0}^{N} L\big(y, F(x, w)\big) \quad (6)$$

The sampling algorithm of GOSS is utilized as LGBM whereas the large gradient have been retained during the sample of small gradient that get selected by providing constant weight. The GOSS is focused majorly on undertrained sample with no change in distribution of raw data. The splitting of instance in variation gain for instance over subset M and N is expressed by equation 7.

$$\hat{V}_j = \frac{1}{n}(L_1 + L_2) \quad (7)$$

Where,
L_1 and L_2 = Variable for subset M and N.

Moreover, the sample with large gradient is represented through M and N illustrates the size selection with discretionarily. In each gradient iteration boosting, the loss function of negative gradient based on the GBDT method output. The principle of LGBM classifier algorithm has not only optimizing the train sample through GOSS but also using EFB for extracting features to increase the network training speed. The data is basically sparse due to high-dimensional data for mutual exclusive features in which the sparse feature are bound together by feature based EFB. This is used for creating recent features and reconstructed based on new features of histogram equations shown in figure 3.

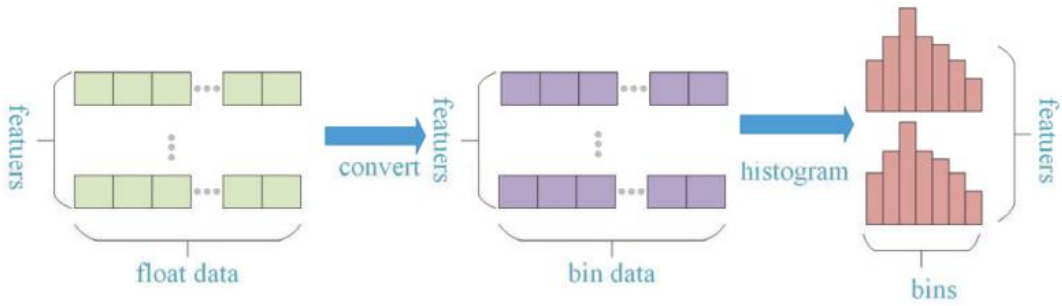

Figure 3 LGBM with histogram based reconstructed features

LightGBM employs the leaf-wise method of growth. In order to prevent fruitless node splitting and conserve computer resources, it can be understood as choosing the most advantageous leaf nodes for growth at each divisive node. In addition, the tree's growth is constrained by the maximum depth, which helps to manage the network's complexity and prevent over-fitting. The generalization capacity of the LightGBM model is also ensured by increasing the network's training speed. Therefore, the AE with LGBM has generated high accuracy through better training of data modeling 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 assist in training the data preprocessed sample that has been split as 70% as train dataset and 30% as test dataset. By importing the lazy predict library in python, the classifier model accuracy is defined and sorted in an ascending order.

EXPERIMENTAL RESULTS

Figure illustrates the pre-trained model of AE is utilized to determine the bias of the model and assist to train classifier model as the fine-tuned model for improving the accuracy of the classifier model in which LGBM has performed better while compared to Extra Tree (ET) classifier and Extreme Gradient Boost (XGBoost) classifier. This experimental research utilizes 500 epochs for better learning and understanding of features. The evaluated outcomes are consistent and linked to different accuracy metrics determined by the LGBM Classification model of the optimal model. The attacks are predicted rapidly by fine-tuning the models.

```
# Train the autoencoder
history = autoencoder.fit(X_train, X_train, epochs=500, batch_size=16, validation_data=(X_test, X_test)

# Extract learned features from the encoder part
encoder_model = keras.Model(inputs=input_layer, outputs=encoder)
deep_learning_features_train = encoder_model.predict(X_train)
deep_learning_features_test = encoder_model.predict(X_test)
```

```
Epoch 1/500
180/180 [==============================] - 1s 3ms/step - loss: 1422976.7500 - val_loss: 68379.7578
Epoch 2/500
180/180 [==============================] - 1s 3ms/step - loss: 17806.2285 - val_loss: 42413.5391
Epoch 3/500
180/180 [==============================] - 0s 3ms/step - loss: 41497.8984 - val_loss: 2462.4470
Epoch 4/500
180/180 [==============================] - 0s 2ms/step - loss: 965.7145 - val_loss: 20768.7559
Epoch 5/500
180/180 [==============================] - 1s 3ms/step - loss: 17740.5742 - val_loss: 2365.1558
Epoch 6/500
180/180 [==============================] - 0s 2ms/step - loss: 1004.9792 - val_loss: 195108.3125
Epoch 7/500
180/180 [==============================] - 0s 2ms/step - loss: 135054.7969 - val_loss: 50624.5039
Epoch 8/500
180/180 [==============================] - 0s 2ms/step - loss: 20529.5645 - val_loss: 80681.1953
Epoch 9/500
180/180 [==============================] - 1s 3ms/step - loss: 239254.6875 - val_loss: 31300.8320
Epoch 10/500
```

Figure 4 Epochs for AE with LSTM classifier

The different performance metrics are precision, recall, f1 score, and support. There are four attributes in the confusion matrix as True Positive (TP) defines the correctly mentioned malware attack is predicted exactly, True Negative (TN) defines the correctly mentioned normal type is predicted exactly. In the case of False Positive (FP), the attack types is correct in actual but wrongly predicted and False Negative (FN) scenario is about attack type is normal but wrongly predicted. Figure 6 illustrate the confusion matrix of the AE with LGBM method in which classes for multivariable is provided and the test dataset involved with the sample of 1235 data transactions.

Table 2 illustrates the confusion matrix class value for top three lazy predict with AE method in which the lable 5 as slowite has obtained the similar value for all three classifier but TP count is high AE with LGBM model while compared with ET and XGBoost classifier.

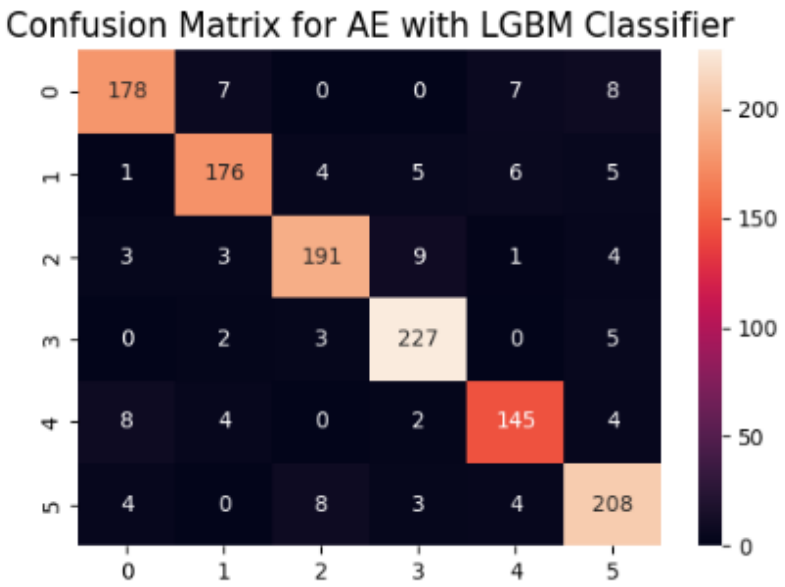

Figure 5 Confusion matrix for AE with LGBM classifier

Table 2 Confusion matrix classes for various ML classification methods

ML classifier methods	Attack type Lable	Confusion Matric classes			
		TP	TN	FP	FN
LGBM Classifier	0	178	1019	22	16
	1	176	1022	21	16
	2	191	1009	20	15
	3	227	979	10	19
	4	145	1054	18	18
	5	208	982	19	26
ET Classifier	0	171	1019	22	23
	1	169	1020	23	23
	2	185	1002	27	21
	3	227	972	17	19
	4	145	1054	18	18
	5	208	978	23	26
XGBoost Classifier	0	171	1019	22	23
	1	169	1016	27	23
	2	185	1002	27	21
	3	217	972	17	29
	4	145	1048	24	18
	5	208	978	23	26

Figure 6 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 which AE with LGBM has high accuracy as 91.09% while compared with ET classifier and XGBoost classifier as 89.47% and 88.66% correspondingly. Similarly, the individual lable weight is measured and estimated through weighed precision, weighted recall and weighted F1-score. According to the experimental results, it determined weighted precision, weighted recall is 91.09% and weighted F1-score is 91.08% respectively which is better in determining the attack type accurately than ET classifier and XGBoost classifier.

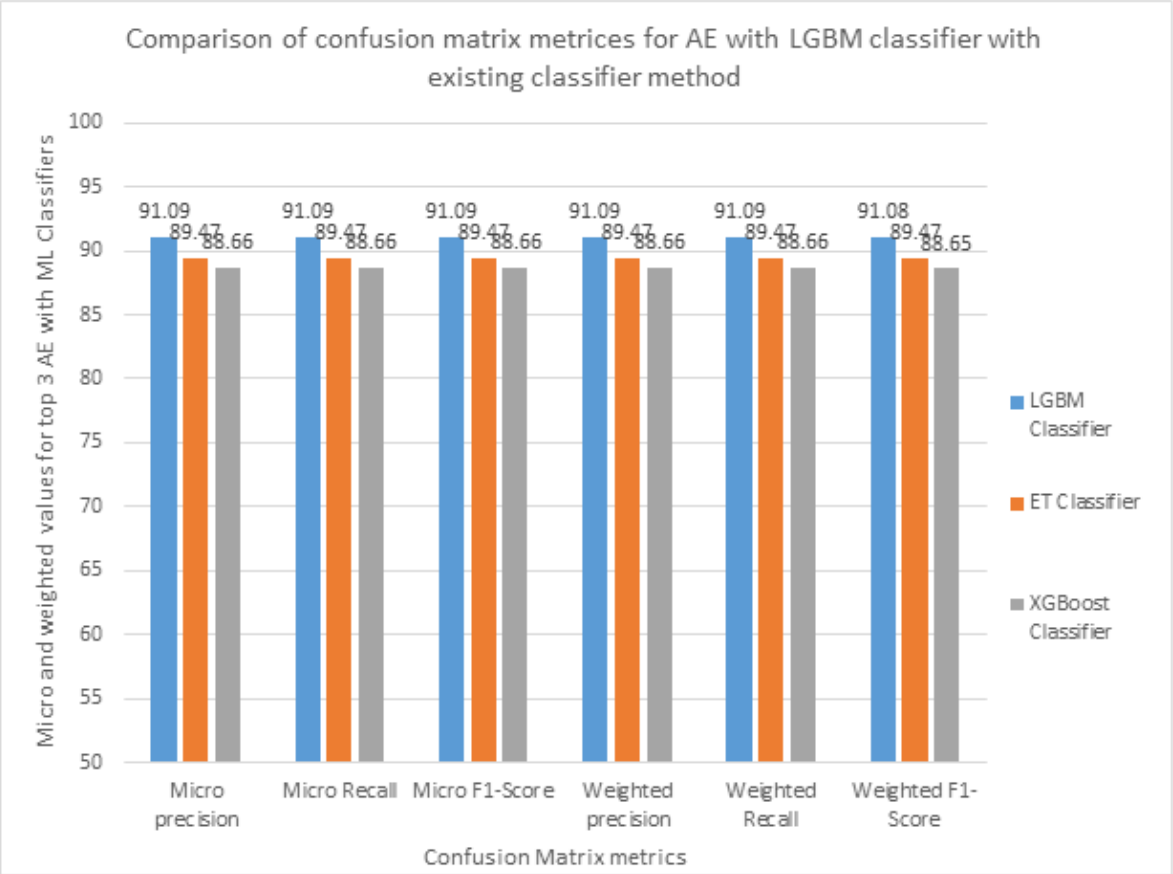

Figure 6 Micro and weighted value for various ML classifiers

CONCLUSION

A robust classification model that can identify active malware attacks based on network flow traffic and benign traffic in SIoT that assist in detecting these attacks early and prevent system tampering. Numerous node compromises might result from the existence of just one rogue node. This proposed AE with LGBM classifier perform as a defensive SIoT from vulnerable nodes after securing data packets and transmission. The LGBM classifier model AE helps in pre-training stage that have a better solution in executed detection of various malware attacks. This research work uses lazy predict classifier to identify attacks like DOS, flood, slowite, brute force and malformed attacks. The parameter analyzed here is precision, recall, f1 score, and support. The accuracy score is 91.09% which determined the best model for malware attack detection in a faster way. Hence, the AE with LGBM classifier has high detection of malware attacks.

REFERENCE

1. J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125–1142, 2017.
2. K. Li, L. Tian, W. Li, G. Luo, and Z. Cai, “Incorporating social interaction into three-party game towards privacy protection in iot,” Computer Networks, vol. 150, pp. 90–101, 2019.
3. W. Meng, K.-K. R. Choo, S. Furnell, A. V. Vasilakos, and C. W. Probst, “Towards bayesian-based trust management for insider attacks in healthcare software-defined networks,” IEEE Transactions on Network and Service Management, vol. 15, no. 2, pp. 761–773, 2018.
4. Panda, G.K., Tripathy, B.K., and Padhi, M.K. 2017. "Evolution of Social Iot World: Security Issues and Research Challenges," in Internet of Things (Iot): Technologies, Applications, Challenges and Solutions. CRC Press, pp. 77-98.
5. Tripathy, B.K., Dutta, D., and Tazivazvino, C. 2016. "On the Research and Development of Social Internet of Things," in: Modeling and Optimization in Science and Technologies. Springer Verlag, pp. 153- 173.
6. Gulati, N., and Kaur, P.D. 2019. "When Things Become Friends: A Semantic Perspective on the Social Internet of Things." Springer Verlag, pp. 149-159.
7. Kuseh SW, Nunoo-Mensah H, Klogo GS, Tchao ET (2022) A survey of trust management schemes for social internet of things. Inform Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi 7(1):48–58.
8. De Oliveira GH, de Souza Batista A, Nogueira M, dos Santos AL (2022) Access control for IoT based on network community perception and social trust against Sybil attacks. Int J Netw Manag 32(1):e2181

9. Sagar S, Mahmood A, Sheng QZ, Pabani JK, Zhang WE (2022) Understanding the trustworthiness management in the social internet of things: a survey. arXiv preprint arXiv:2202.03624.
10. Ning Z, Zhou MC, Yuan Y, Ngai EC, Kwok RYK (2022) Guest editorial special issue on collaborative edge computing for social internet of things systems. IEEE Trans Comput Soc Syst 9(1):59–63.
11. R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," IEEE transactions on dependable and secure computing, vol. 13, no. 6, pp. 684–696, 2015.
12. K. Zhao and L. Pan, "A machine learning based trust evaluation framework for online social networks," in 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2014, pp. 69–74.
13. A. M. Kowshalya and M. Valarmathi, "Trust management for reliable decision making among social objects in the social internet of things," IET Networks, vol. 6, no. 4, pp. 75–80, 2017.
14. B. Jafarian, N. Yazdani, and M. S. Haghighi, "Discriminative-aware trust management for social internet of things," Computer Networks, p. 107254, 2020.
15. R. Chen, J. Guo, D.-C. Wang, J. J. Tsai, H. Al-Hamadi, and I. You, "Trust-based service management for mobile cloud iot systems," IEEE transactions on network and service management, vol. 16, no. 1, pp. 246–263, 2018.
16. L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the internet of things," IEEE Communications Magazine, vol. 52, no. 1, pp. 97–105, jan 2014.
17. 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.
18. W. Wang, Y. Li, X. Wang, J. Liu, and X. Zhang, ''Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers,'' Future Gener. Comput. Syst., vol. 78, pp. 987–994, Jan. 2018.
19. A. Kumar, K. S. Kuppusamy, and G. Aghila, ''A learning model to detect maliciousness of portable executable using integrated feature set,'' J. King Saud Univ. Comput. Inf. Sci., vol. 31, no. 2, pp. 252–265, Apr. 2019.
20. W. Wang, M. Zhao, and J. Wang, "Effective Android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," J. Ambient Intell. Humaniz. Comput., vol. 0, no. 0, pp. 1–9, 2018.
21. Zhang, Yi, Yuexiang Yang, and Xiaolei Wang. "A Novel Android Malware Detection Approach Based on Convolutional Neural Network." Proceedings of the 2nd International Conference on Cryptography, Security, and Privacy. ACM, 2018.