Classification of Iot Network Traffic Using Machine Learning Algorithms

R. Kavitha, P. Saranya

Department of Computer Science and Engineering, Rohini College of Engineering and Technology, Anna University, Chennai

Assistant Professor, Department of Computer Science and Engineering, Rohini College of Engineering and Technology, Anna University, Chennai

Abstract-The identification of IoT devices in network data is one such essential operation. It enables the administrator to keep monitors on the actions of IoT devices, which can be helpful for the effective implementation of Quality of Service, detect malicious IoT devices, etc. In this study, a machine learning based classification of IoT network traffic is proposed. IoT traffic classification is separated using a larger data set. The input dataset is first pre-processed to remove any noise. Chi-square-based feature extraction is used during the extraction process. The Chi-Square technique is used to process the extraction areas in order to extract various features and choose the necessary features in order to improve classification. The KNN and MLP classifiers are employed for determine the precise classification. The output of the proposed technique is implemented by using the Python software. As a result this approach achieves good accuracy but takes large training times in packet level due to large amounts of data and unbalanced data.

I. INTRODUCTION

IoT Networks are the networks of connected devices, which uses to share or spread data to other devices or interfaces that are accessible [1]. For IoT sensors and devices to communicate, there are numerous types of IoT networks are exist. In the past, IoT technology is only used in homes and small workplaces; but, today, IoT technology is being incorporated into various businesses for increased reliability and time savings. There has been a recent increase in traffic as a result of the development of Internet of Things (IoT) applications. These intelligent objects include devices, instruments, cars, structures, and other things, which have electronic circuits, sensors, software as well as network connectivity integrated into it [2-5]. In the Internet of Things, all active nodes are managed remotely by means of network infrastructure. In today's world, every system places a high value on network security and service quality. In order to do this, everyone has to be categories the IoT network traffic. This will allow users to efficiently analyse the vast amounts of data [6-7]. IoT traffic include a variety of data flows which are sent back and forth between devices, as well as sources for traffic attacks and a considerably bigger volume of data. Consequently, this traffic classification process is quite difficult. As a result, even though various prediction algorithms have been developed, network traffic prediction is critical from a variety of perspectives. The end-to-end network traffic in IoTs has been proposed, but it remains an insurmountable barrier. Traffic on the network currently. The cost of the algorithm and accuracy are the main trade-offs, which prediction systems consider [8-10].

In response to the aforementioned difficulties, this paper concentrate on the issue of end to end network traffic prediction in the IoTs and suggest a technique based on Machine learning (ML) for real-time traffic forecasting [11]. Machine learning provides solutions to the complicated problems of today's reality.

It is suitable for IoT-based conditions where it is very difficult to organize the traffic and to predict the traffic load due to vast data, which machine learning algorithms has gradually modify as well as enhance from comprehension. For network management, machine learning-based network traffic prediction consumes a lot of compute and memory resources. So that these algorithms are train on a prior sample of network traffic [12-13].The IoT organization is overloaded by unfavourable or retaliatory information. Thus, it is necessary to properly split the traffic, hold up the abnormal traffic as well as assign the regular traffic to the appropriate channels. The conventional method is integrated with a machine learning strategy to enhance the performance of IoT in heterogeneous networks, which is based on a PCA-based feature extraction method. Heterogeneous data have their dimensions reduced using PCA without losing any of their original properties. As a result, the feature extraction shortens the machine learning training process and uses less power. Therefore this study use the Chi-square Method is used to carry out the feature extraction process [14-15].

In this study, machine learning based classification of IoT network traffic is proposed. The data preprocessing is carried out to eliminate the noise obtained from the dataset. Chi-Square approach is employed to carry out the feature selection. KNN and MLP classification is employed to predict the accurate output. Model evaluation is used to select the best algorithm, which suitable for the dataset for calculating a particular issues. The output of the results is implemented in the Python software. As a result the suggested technique achieves good accuracy but takes large training times in packet level due to large amounts of data and unbalanced data.

II. PROPOSED METHODOLOGY

The dataset is fed as an input, the data pre-processing is used to remove the noise from the dataset. After that it is sent to the chi-square method for feature selection. Categorical features in a dataset are tested using the chi-square method. The KNN and MLP classifiers are effective for classifying the data and the results, which such method predict are reliable. Model evaluation is the process of analysing a machine learning model's performance and identifying its advantages and disadvantages using various evaluation measures.

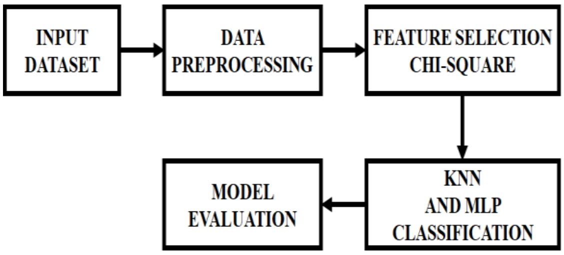


Figure 1 Block diagram of the proposed wok

A) DATA PREPROCESSING

Data preprocessing, a part of data preparation, refers to any type of processing done on raw data to get it ready for another data processing technique. It has historically been a significant first step in the data mining process. Data preparation is necessary because good data is obviously more essential than good models and the quality of the data is of utmost importance. As a result, businesses and individuals devote a significant amount of time to cleaning and preparing the data for modelling. The data which is currently available in the actual world is highly noisy, incorrect, and incomplete. It lack pertinent, specific attributes, have missing data, or even false and inaccurate values. The preparation of data has to be improved in quality. By removing any duplicates or anomalies, normalizing the data for comparison, and enhancing the accuracy of the outputs, preprocessing helps to make the data consistent.

Machine learning data preparation methods can be broadly divided into two categories:

- Data Cleaning
- Data Transformation

a) Data cleaning

Data cleaning is the process of repairing or eradicating inaccurate, corrupted, improperly formatted, duplicate, or insufficient data from a dataset. Data duplication or labelling errors are common when merging various data sources. The data from the real world may not be precise, consistent, right, and pertinent. Getting the data clean is the first and most important step. In this stage, there are several steps, including such as,

- The characteristics not match the data dictionaries and have the wrong data types. Before starting any form of data cleansing, the data types has to be corrected.
- Check for negative values as well as null or missing values. The data will determine whether the negative numbers are relevant. A negative value in the income column is fictitious, but the identical negative value in the profit column turns into a loss.
- Aligning the date column's format with the data analysis tool's format.

b) Data transformation

Data transformation is the technical process of translating data from one format, standard, or structure to another without altering the dataset's content. This is often done to make the data more usable by users or apps or to enhance the quality of the data.

B) FEATURE SELECTION CHI-SQUARE

The process of collecting the most pertinent characteristics from the dataset and then using machine learning methods to improve the performance of the model is known as feature selection. A huge number of pointless features exponentially lengthens training time and raises the possibility of overfitting. Feature categories in a dataset are tested using the chi-square method. Select the required number of features with the best Chi-square scores by computing the Chi-square between each feature and the objective. It examines whether the sample's association between two categorical variables accurately reflects that association in the general population. An approach to selecting the input features thought to be most helpful to a model in predicting the target variable is known as supervised feature selection. It use either filter based techniques or wrapper techniques for the supervised feature selection technique. The best features are chosen using a wrapper-based method, such as Recursive Feature Elimination (RFE). However, it used the filter-based feature selection strategy, which scores the relationship among the features as well as the target labels, or class labels, to choose the features from our feature space F. Because the goal class labels are category in nature and input features are numerical or become quantitative after preprocessing, it specifically chose the ANOVA (Analysis of Variance) F-value feature selection method.

C) KNN AND MLP CLASSIFICATION

One of the most basic types of machine learning algorithms, KNN is mostly used for categorization. The classification is based on how the neighboring data point is categorized. KNN categorizes the newly added data points depending on how similar they are to the previously stored data points. The parameter "k" in the KNN system denotes the number of nearest neighbours to be included in the voting process. KNN determines the distances among a query as well as each example in the data, chooses the K examples closest to the query and then votes for the label with the highest frequency or averages the labels.

Based on the presumption the objects with comparable neighbours will have similar prediction values, the nearest neighbour technique for prediction is used. Finding the k points in the multidimensional space Rn which are closest to the unknown sample as well as classifying the unknown sample based on the k points' categories is the basic principle of the nearest neighbour algorithm. These k points are the unknown samples' k closest neighbours. All instances are taken to be points in dimensional space by this method. According to the usual Euclidean distance, the closest neighbour of an instance is determined. The eigenvector of x given as follows:

$$<a_1(X), a_2(X), \dots, a_n(X)> \quad (1)$$

Where, a_r(x) Represents the rth attribute value of instance x.

The distance between the two instances X_i and X_j is defined as$d(X_i, Y_j)$,

$$d(X_i, Y_j) = \sqrt{\sum_{r=1}^{n}\left(ar(Xi) - ar(xj)\right)2} \quad (2)$$

Utilizing 1-of-C coding is a popular method for creating an MLP for a C-class classification task. The MLP has C outputs using this coding method, where each output relates to a single class. Using the back-propagation approach, the network is trained to produce a high value at the output that corresponds to the right class while the other outputs are low. An MLP with a single hidden layer's cth output, y$_c$, can be written as

$$y_c = f_o\left(\sum_{j=0}^{M_H} w_{cj}\, f_h\left(\sum_{i=0}^{M_I} w_{ji}\, x_i\right)\right) \quad c = 1, \dots, C \quad (3)$$

Here, x_i specifies the ith input, f_h, f_o indicates the sigmoidal activation functions for the hidden as well as output layers,w_{ji}, w_{cj}represents the input and output to hidden weights, if the f_o is logistic function,

$$f_o(\emptyset) = \frac{1}{1+e^{-K\emptyset}} \quad (4)$$

Where, K specifies the constant, It is possible to interpret the MLP outputs as probability posteriors. However, there is no guarantee that the outputs entirety because there is a chance that the outputs with a finite training data set, which not precisely predict the posteriors. However, the results are unreliable when the MLP is given non-class data.

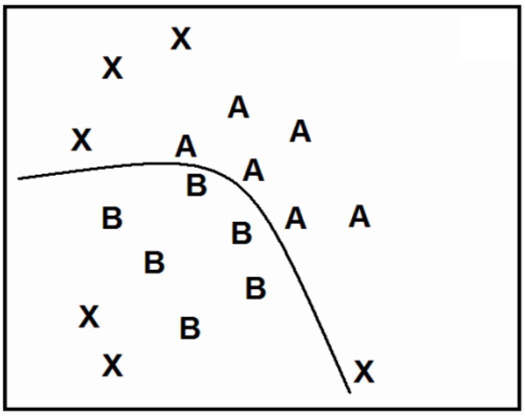


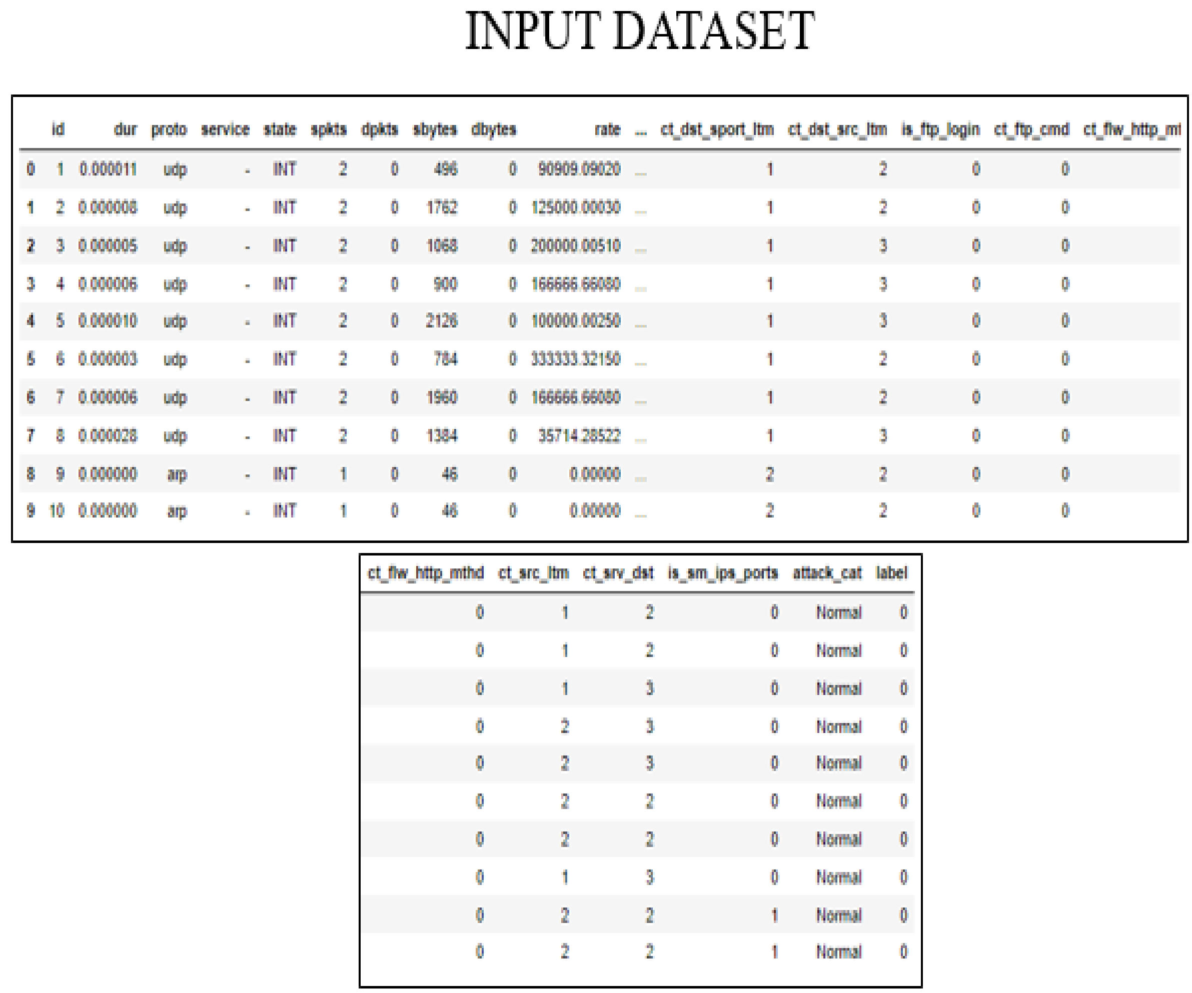
Figure 2 Decision boundaries formed by training MLP classification

D) MODEL EVALUATION

By using this evaluation technique, we can determine which algorithm will work best to solve a specific problem using the provided dataset. Similar to this, "Best Fit" is the term used in machine learning. It compares the effectiveness of various machine learning models using the same input dataset. The technique of evaluation places a strong emphasis on the model's ability to accurately anticipate final results. Select the algorithm which offers greater accuracy for the input data as well as it is regarded as the best model since it more accurately predicts the outcome out of the various algorithms everyone utilize in the stage. When employing machine learning to address various issues, accuracy is regarded as the primary consideration. If the accuracy is high, the model's predictions based on the provided data will also be accurate to the maximum extent possible. An ML problem is solved in various stages, including data collection, problem definition, brain storming with the available data, preprocessing, transformation, evaluation as well as evaluation. Although an ML model goes through numerous steps, the evaluation stage is the most important since it allows us to gauge how accurately the model predicts the future. The final decision about the effectiveness and use of the ML model is made in terms of accuracy metrics.

III. RESULTS AND DISCUSSION

A machine learning based classification of IoT network traffic is suggested. With more than 99% accuracy, everyone identified IoT devices by using a multi stage machine learning based categorization system. The effectiveness of those machine learning algorithms in terms of classification precision, processing speed, training time as well as other factors. Moreover, a few recommendations for choosing the machine learning algorithm for different use cases are provided based on the outcomes.



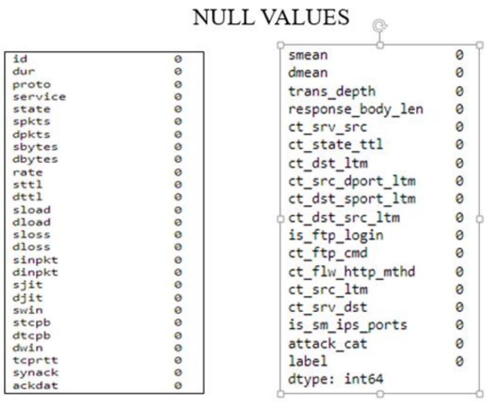
INPUT DATASET

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mt
0	1	0.000011	udp	-	INT	2	0	496	0	90909.09020	...	1	2	0	0	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.00030	...	1	2	0	0	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.00510	...	1	3	0	0	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.66080	...	1	3	0	0	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.00250	...	1	3	0	0	
5	6	0.000003	udp	-	INT	2	0	784	0	333333.32150	...	1	2	0	0	
6	7	0.000006	udp	-	INT	2	0	1960	0	166666.66080	...	1	2	0	0	
7	8	0.000028	udp	-	INT	2	0	1384	0	35714.28522	...	1	3	0	0	
8	9	0.000000	arp	-	INT	1	0	46	0	0.00000	...	2	2	0	0	
9	10	0.000000	arp	-	INT	1	0	46	0	0.00000	...	2	2	0	0	

ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	attack_cat	label
0	1	2	0	Normal	0
0	1	2	0	Normal	0
0	1	3	0	Normal	0
0	2	3	0	Normal	0
0	2	3	0	Normal	0
0	2	2	0	Normal	0
0	2	2	0	Normal	0
0	1	3	0	Normal	0
0	2	2	1	Normal	0
0	2	2	1	Normal	0

Figure 3 Input dataset

The input dataset is represented in figure 3, which illustrates the Id, proto, state, spkts, dpkts, sbytes, and dbytes etc, which are displayed in the page.



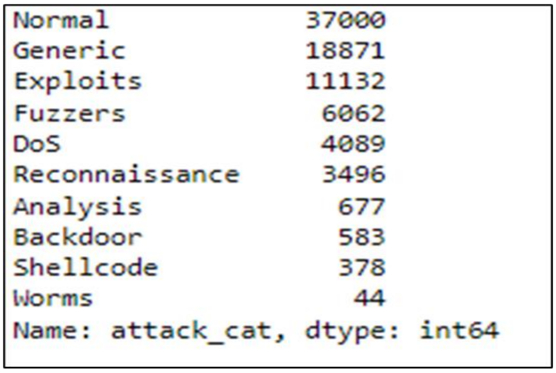
NULL VALUES

```
id          0
dur         0
proto       0
service     0
state       0
spkts       0
dpkts       0
sbytes      0
dbytes      0
rate        0
sttl        0
dttl        0
sload       0
dload       0
sloss       0
dloss       0
sinpkt      0
dinpkt      0
sjit        0
djit        0
swin        0
stcpb       0
dtcpb       0
dwin        0
tcprtt      0
synack      0
ackdat      0
```

```
smean                0
dmean                0
trans_depth          0
response_body_len    0
ct_srv_src           0
ct_state_ttl         0
ct_dst_ltm           0
ct_src_dport_ltm     0
ct_dst_sport_ltm     0
ct_dst_src_ltm       0
is_ftp_login         0
ct_ftp_cmd           0
ct_flw_http_mthd     0
ct_src_ltm           0
ct_srv_dst           0
is_sm_ips_ports      0
attack_cat           0
label                0
dtype: int64
```

Figure 4 Null values

Figure 4 represents the preprocessing techniques. In this technique is used to find the null values and repeated values in the data set.



```
Normal            37000
Generic           18871
Exploits          11132
Fuzzers            6062
DoS                4089
Reconnaissance     3496
Analysis            677
Backdoor            583
Shellcode           378
Worms                44
Name: attack_cat, dtype: int64
```

Figure 5 Attacks

Figure 5 specifies the name like attack_cat, dtype: int 64 are displayed in the page.

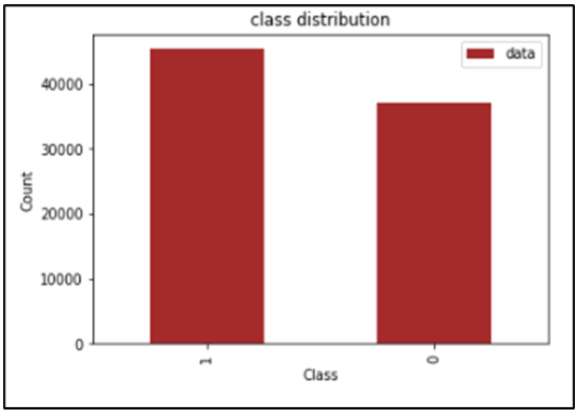


Figure 6 Class distribution

A class distribution can be defined as a dictionary where the key is the class value (e.g. 0 or 1) and the value is the number of randomly generated examples to include in the dataset.

The class distribution is specified that the class is differ with the count.

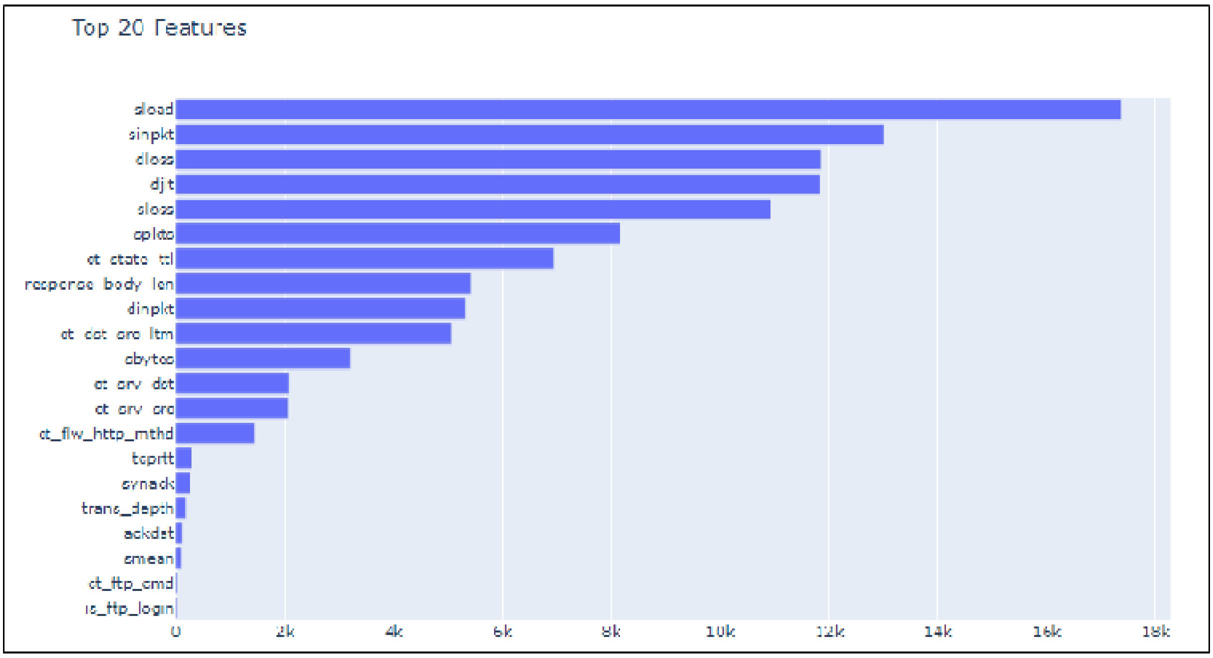
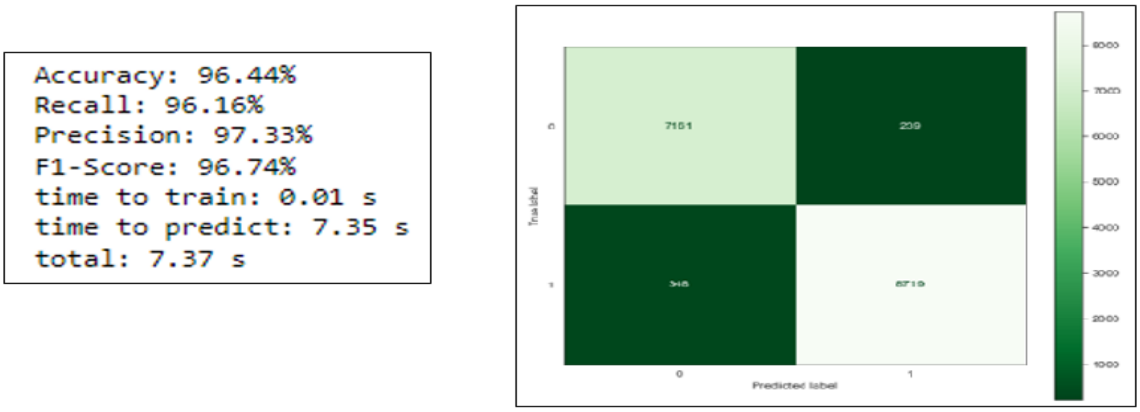


Figure 7 Top features

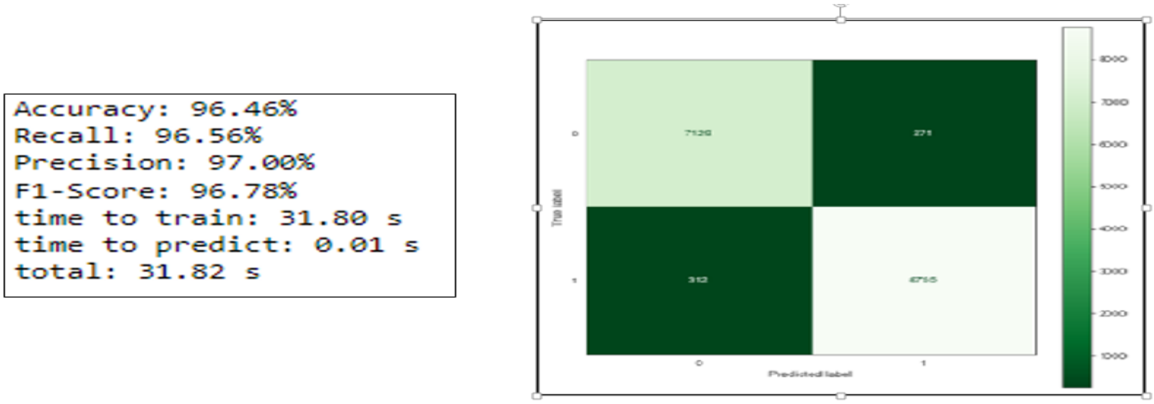
Figure 7 specifies the top features, which is analyzed that the top 20 features are displayed in the page that shown the features are varies from one another.



```
Accuracy: 96.44%
Recall: 96.16%
Precision: 97.33%
F1-Score: 96.74%
time to train: 0.01 s
time to predict: 7.35 s
total: 7.37 s
```

Figure 8 output of KNN

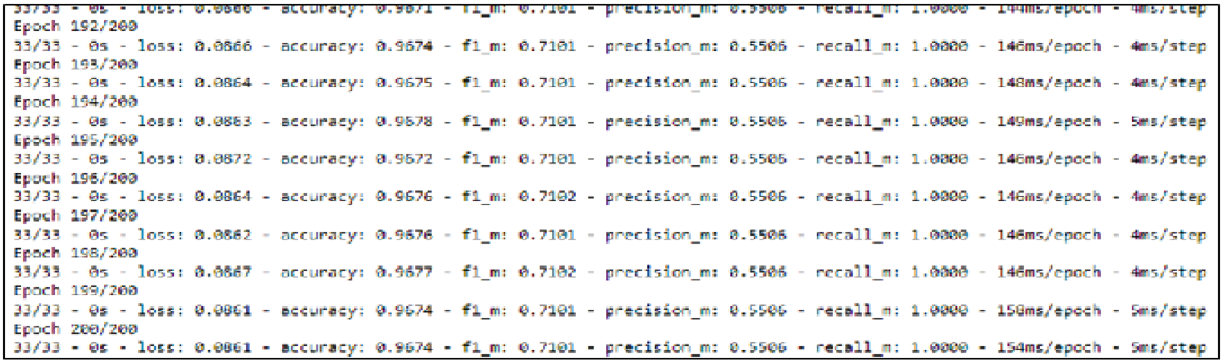
The output of KNN is represented in figure 6, from the figure it is observed that the accuracy attains 96.44%, recall 96.16%, precision 97.33%, F1-score 96.74%, time to train 0.01s, time to predict 7.35s and total 7.37s.



```
Accuracy: 96.46%
Recall: 96.56%
Precision: 97.00%
F1-Score: 96.78%
time to train: 31.80 s
time to predict: 0.01 s
total: 31.82 s
```

Figure 9 Output of MLP

Figure 9 illustrates the output of MLP, which is analyzed that the MLP attains the accuracy level of 96.46%, recall 96.56%, precision 97.00%, time to train 31.80s, time to predict 0.01s and total 31.82s as shown in above figure.



```
33/33 - 0s - loss: 0.0866 - accuracy: 0.9671 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 144ms/epoch - 4ms/step
Epoch 192/200
33/33 - 0s - loss: 0.0866 - accuracy: 0.9674 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 146ms/epoch - 4ms/step
Epoch 193/200
33/33 - 0s - loss: 0.0864 - accuracy: 0.9675 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 148ms/epoch - 4ms/step
Epoch 194/200
33/33 - 0s - loss: 0.0863 - accuracy: 0.9678 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 149ms/epoch - 5ms/step
Epoch 195/200
33/33 - 0s - loss: 0.0872 - accuracy: 0.9672 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 146ms/epoch - 4ms/step
Epoch 196/200
33/33 - 0s - loss: 0.0864 - accuracy: 0.9676 - f1_m: 0.7102 - precision_m: 0.5506 - recall_m: 1.0000 - 146ms/epoch - 4ms/step
Epoch 197/200
33/33 - 0s - loss: 0.0862 - accuracy: 0.9676 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 146ms/epoch - 4ms/step
Epoch 198/200
33/33 - 0s - loss: 0.0867 - accuracy: 0.9677 - f1_m: 0.7102 - precision_m: 0.5506 - recall_m: 1.0000 - 146ms/epoch - 4ms/step
Epoch 199/200
33/33 - 0s - loss: 0.0861 - accuracy: 0.9674 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 150ms/epoch - 5ms/step
Epoch 200/200
33/33 - 0s - loss: 0.0861 - accuracy: 0.9674 - f1_m: 0.7101 - precision_m: 0.5506 - recall_m: 1.0000 - 154ms/epoch - 5ms/step
```

Figure 10 Model performance

The model performance of the proposed technique is represented in figure 8, which is observed that the output of accuracy, recall, precision, F-1 score, time to train, time to predict and total time is specified for KNN and MLP

IV CONCLUSION

A machine learning-based classification of IoT network traffic is proposed in this study. Despite the widespread use of IoT devices in smart homes, businesses, campuses, and cities worldwide, it is difficult for operators of these environments to see which IoT devices are connected to their networks, how much traffic they generate, and whether or not the devices are operating properly and without security flaws. To characterization and classification of IoT devices the suggested method instrumented a smart environment with 28 different IoT devices over the period of 26 weeks as well as continuously gathered traffic traces. Traffic is described by using cypher suites, activity cycles, signaling patterns as well as communication protocols. This method developed a multi-stage machine learning-based categorization system that accurately categorizes IoT devices with 99% or more of the time. The effectiveness of those machine learning algorithms in terms of classification precision, processing speed training time as well as other factors. Finally, a few suggestions for choosing the machine learning algorithm for different use cases were provided based on the results.

REFERENCE

[1] Shafiq, Muhammad, Zhihong Tian, Ali Kashif Bashir, Xiaojiang Du, and Mohsen Guizani. "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques." *IEEE Internet of Things Journal* 8, no. 5 (2020): 3242-3254.

[2] Nižetić, Sandro, Petar Šolić, Diego López-de-Ipiña González-De, and Luigi Patrono. "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future." *Journal of Cleaner Production* 274 (2020): 122877.

[3] Sethi, Pallavi, and Smruti R. Sarangi. "Internet of things: architectures, protocols, and applications." *Journal of Electrical and Computer Engineering* 2017 (2017).

[4] Rock, Leong Yee, Farzana Parveen Tajudeen, and Yeong Wai Chung. "Usage and impact of the internet-of-things-based smart home technology: a quality-of-life perspective." *Universal Access in the Information Society* (2022): 1-20.

[5] Lin, Huichen, and Neil W. Bergmann. "IoT privacy and security challenges for smart home environments." *Information* 7, no. 3 (2016): 44.

[6] Shouran, Zaied, Ahmad Ashari, and Tri Priyambodo. "Internet of things (IoT) of smart home: privacy and security." *International Journal of Computer Applications* 182, no. 39 (2019): 3-8.

[7] Khedkar, Shilpa P., R. Aroul Canessane, and Moslem Lari Najafi. "Prediction of traffic generated by IoT devices using statistical learning time series algorithms." *Wireless Communications and Mobile Computing* 2021 (2021): 1-12.

[8] Kumar, Rakesh, Mayank Swarnkar, Gaurav Singal, and Neeraj Kumar. "Iot network traffic classification using machine learning algorithms: an experimental analysis." *IEEE Internet of Things Journal* 9, no. 2 (2021): 989-1008.

[9] Kumar, Rakesh, Mayank Swarnkar, Gaurav Singal, and Neeraj Kumar. "Iot network traffic classification using machine learning algorithms: an experimental analysis." *IEEE Internet of Things Journal* 9, no. 2 (2021): 989-1008.

[10] Umair, Muhammad Basit, Zeshan Iqbal, Muhammad Bilal, Tarik Adnan Almohamad, Jamel Nebhen, and Raja Majid Mehmood. "An efficient internet traffic classification system using deep learning for IoT." *arXiv preprint arXiv: 2107.12193* (2021).

[11] Meidan, Yair, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. "ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis." In *Proceedings of the symposium on applied computing*, pp. 506-509. 2017.

[12] Han, Shangbin, Qianhong Wu, and Yang Yang. "Machine learning for Internet of things anomaly detection under low-quality data." *International Journal of Distributed Sensor Networks* 18, no. 10 (2022): 15501329221133765.

[13] Oha, Chibueze Victor, Fathima Shakoora Farouk, Pujan Pankaj Patel, Prithvi Meka, Sowmya Nekkanti, Bhageerath Nayini, Smit Xavier Carvalho, Nisarg Desai, Manishkumar Patel, and Sergey Butakov. "Machine learning models for malicious traffic detection in IoT networks/IoT-23 dataset." In *International Conference on Machine Learning for Networking*, pp. 69-84. Cham: Springer International Publishing, 2021.

[14] R. Kalakoti, S. Nõmm and H. Bahsi, "In-Depth Feature Selection for the Statistical Machine Learning-Based Botnet Detection in IoT Networks," in IEEE Access, vol. 10, pp. 94518-94535, 2022.

[15] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021.