ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

Article Title: IoT Network Traffic Classification using Deep Learning Algorithms

IoT Network Traffic Classification using Deep Learning Algorithms

A. Somasundaram[1,*]

[1]Assistant Professor, Department of Electrical and Electronics Engineering, St. Mother Theresa Engineering College. somasundaram4ever@gmail.com

ABSTRACT

This project suggests utilizing a deep learning system to classify IoT networks. The preprocessing technique receives the input data. The dataset's null and repeated values are eliminated using this method. Following pre-processing, the preprocessed data is moved into the feature selection block. The feature selection process makes use of the chi square. The extraction areas are processed using the Chi-Square technique to extract different characteristics and select the aspects that are needed to improve classification. Subsequently, the data extraction procedure moves on to categorization. Convolutional neural networks (CNNs) are utilized for classification based on the model's performance analysis using CNN classification. This paper provides a useful model to classify IoT network traffic in an actual IoT scenario. The software used to implement this project is Python.

Keywords: Convolutional neural networks (CNNs), Recursive Feature Elimination (RFE), Internet Traffic Classification (ITC), Deep Learning (DL), Neural Networks (NNs), Machine Learning (ML).

1 Introduction

In order to enhance network performance, intrusion detection, QoS provisioning, and dynamic access control, internet traffic classification, or ITC, is essential. As a result, a number of ITC strategies, including the machine-learning-based classification system, payload-based classification method, and port-based classification method, are being developed recently [1]. IoT networks' traffic patterns show how they differ from traditional networking approaches. Specifically, very homogeneous traffic which occurs when multiple devices running the same application show similar behavior patterns is what distinguishes the IOT [2]. Furthermore, because IOT networks consist of numerous devices, they produce enormous volumes of information. All of the network's devices are susceptible to an assault that modifies the network's usual behavior and poses serious issues for end users. It is crucial to find these network anomalies as soon as possible because this is a pressing matter. Many applications are undergoing extensive development of techniques to detect anomalous activity [3]. Some techniques, such protocol creation, network administration, and resource distribution, are predicated on the analysis of network traffic characteristics that are necessary to comprehend particular traffic patterns. This process can take a long time, restrict the amount of real-time device monitoring, and call for a high level of skill and understanding. As a result, this becomes problematic when it becomes imperative to promptly handle any possible irregularities in

traffic. Machine learning (ML) has gradually replaced human-assisted traditional intrusion detection systems as a method in recent years [4]. The primary explanation is that ML makes it possible to create portable algorithms. Supervisory, unsupervised, and semi-supervised are the three groups into which machine learning algorithms fall. In supervised learning, training takes place on a labelled dataset. The method of semi-supervised learning utilizes either identified or uncontrolled information. Unauthorized education relies on the classification of unlabeled data [5], where classes are not known. Traffic classification has made use of a variety of machine learning techniques, including SVM, Back propagation, KNNs, and C4.5 [6]. It is imperative that security systems of the future be capable of providing immediate terms identifying anomalies and information base adaptation in response to regular variations in internet activity.

Consequently, ML algorithms ought to be strengthened over time in accordance with modifications in the active utilization of IoT devices. Given the limited memory of end devices, it can be difficult to store all of the data related to their network traffic [7]. Because of this, it is occasionally required to determine if specific network traffic examples is not able to be included in the machine learning technique. In the preceding research, they conduct a thorough investigation of IoT traffic classification by assessing the effectiveness of three Deep Learning (DL) Neural Networks (NNs) and four ML algorithms across various sampling strategies. Our method also looks into how many layers, or NN depth, are required for effective traffic classification. These days, most protocols use a dynamic port allocation mechanism for security concerns [8]. Thus, the application of port-based traffic classification is restricted by the assigning of ports dynamically. Therefore, the payload-based approach was devised to get over these limitations. By examining the packet payload and comparing it to the signature of the protocols contained in the database, this technique classifies packets based on their application-layer information. Nevertheless, putting this strategy into practice is difficult. For instance, the DPI is unable to inspect the packet payload when encrypted traffic is present. Furthermore, from the standpoint of the user, DPI may result in privacy-related problems. The main factor driving the usage of ML and DL algorithms is the massive volumes of data produced by IoT devices [9]. As such, network traffic analysis necessitates technology that can quickly process, analyze and understand large amounts of data. In light of the widespread usage of ML in research pertaining to trending topics, they concentrate on utilizing the most popular algorithms to identify patterns of abnormal traffic [10]. These include the entire system activity information along with arbitrarily under sampled networking flow samples are used for developing the machines, our suggested pipeline successfully classifies anomalous activity. In contrast to earlier research, we examine the effects on the models' performance of varying levels of arbitrary minimizing (simulating train data loss through various sampling procedures). The method extracts network traffic features using a CNN and then applies supervised learning

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

to identify intrusions. DNN-based IDS techniques modify and normalize data before supplying it to the model. .

2 Recent Works

Rajesh Kalakoti et al [2022] [11] have proposed an overview of in-depth feature selection for botnet detection in IoT networks using statistical machine learning. This work focuses on feature set minimization for machine learning tasks, which are divided into six different binary and multiclass classification problems based on the stages of the botnet life cycle. More specifically, they obtained the optimum feature sets for each categorization assignment by combining specific machine learning techniques with filter and wrapper approaches. The feature selection methods favour channel-based features for post-attack, communication, and control stages of detection, even if host-based features are more significant in identifying bot-based attacks. However, the Pearson correlation method finds that 33 characteristics are not significantly correlated.

Houda Jmila et al [2022] [12] have created a survey on the classification of IoT devices in smart homes using machine learning and network traffic analysis. In order to comprehend these methods' potential and constraints, this paper examines them. It begins by outlining a general procedure for classifying Internet of Things devices. After that, it examines the approaches and fixes for every workflow step. Presenting the analysis's conclusions in taxonomies together with statistics illustrating patterns in the literature. Additionally, this study investigates and recommends unexplored or understudied research avenues. Nevertheless, due to the configuration stage might not occur more than once during an IoT device's life cycle, setup traffic is uncommon, sparse, and difficult to collect in real-world network monitoring.

Amir Mahdi Sadeghzadeh et al [2021] [13] have presented a test of the ability of DL-based network traffic classifiers to withstand adversarial network traffic (ANT). By employing Universal Adversarial Perturbation (UAP) generating techniques, ANT leads DL-based network traffic classifiers to make inaccurate predictions. ANT is created in real time since network communication does not need to be buffered before transmission. Three categories should be created out of the input space for the DL-based network traffic classification: packet classification, flow content classification, and flow time series classification. We suggest three additional approaches that insert UAP into network traffic in order to produce ANT. Findings show that adding a small amount of UAP to network data significantly reduces the effectiveness of DL-based network traffic classifiers across the board. The recall of other classes is, however, mostly unaffected by the port attack.

Elaiyasuriyan Ganesan et al [2021] [14] I have introduced a network traffic classification using supervised machine learning models in an SDN-enabled FiWi-IoT smart environment. In order to improve SDN enhanced-FiWi-IoT, this study offers a machine learning supervised network traffic categorization scheduling model that can intelligently learn and guarantee traffic depending on its QoS requirements (QoS-Mapping). Gather the various network traffic

trace files from IoT and non-IoT devices according to traffic flow, then examine the traffic traces to extract statistical information (such as IP addresses, source and destination ports, etc.). ML versions are thought to be inexpensive and lightweight. It is simple to evaluate encrypted traffic. It is necessary to have very expensive hardware for the payload method of search. The signature format of new apps must be updated continuously when using payload-based techniques.

Meenaxi M Raikar et al [2020] [15] have implemented the practice using supervised learning to classify data traffic in software-defined networks (SDN). This paper suggests combining an SDN (software-defined network) architecture with machine learning technology. A software-defined network platform uses three different supervised learning models to classify data flow based on its intended applications: Support Vector Machine (SVM), Nearest Centroid, and Naïve Bayes (NB). After acquiring network traffic traces, the classifier uses the attributes it has developed to create predictions. The accuracy of traffic classification in all three supervised learning models exceeds 90%. It is not possible for the DPI mechanism to categorize encrypted traffic.

3 Proposed Work Explanation

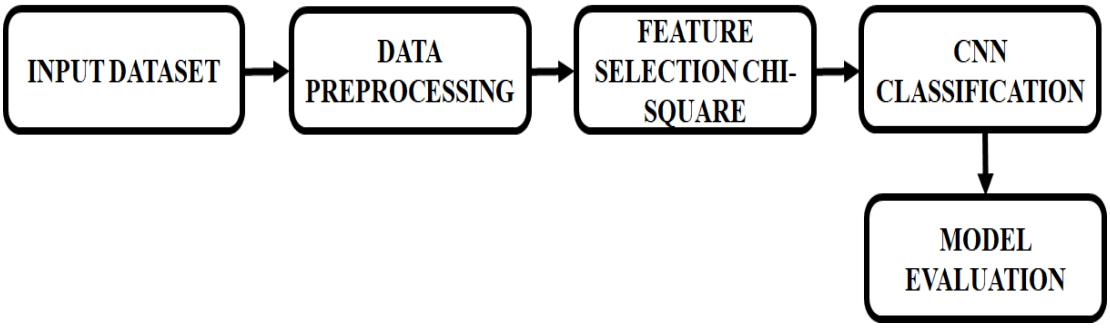

Figure 1: *Proposed Block diagram*

This project suggests utilizing deep learning techniques to classify IoT network traffic. Preprocessing is where the input dataset is relocated. Null and repetitive values in the input dataset are eliminated using this method. Following preprocessing, the preprocessed data is inserted into the block designated for feature selection. The selection of features is done using the chi square. Utilizing the Chi-Square approach, the extraction areas are processed in order to extract various features and choose the essential elements for more accurate categorization. After extraction, the data are sent to the classification phase. Classification is accomplished using convolutional neural networks, or CNNs, by using CNN classification to analyze the model's performance. This project offers a helpful classification model for Internet of Things network traffic that may be implemented in an actual IoT setting. This project is being carried out with Python software.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

3.1 Data Preprocessing.

Preprocessing is the process of applying a series of steps to modify or alter data. It is a modification that we apply to our data before feeding it into an algorithm. The process of acting on data to extract, convert, or update it is known as data processing, especially when carried out by a computer. It is a method for turning dirty data sets into clean ones. Data preprocessing, a particular form of statistical preparation, is any form of analysis applied to initial data to get it ready for an additional analysis method. It has consistently existed a crucial initial step in the data mining procedure. Preparing the data is important because high-quality data is crucial and definitely more important than well-designed models. Because of this, companies and individuals have to put in a lot of work to clean up and prepare the data for modeling. The present corpus of real-world data is noisy, unreliable, incomplete, and has multiple quality issues. It can contain erroneous and inaccurate data, missing numbers, or a lack of exact and pertinent features. The idea of the internet of things is still evolving. Its basic idea is to use the appropriate information and communication technologies to connect analog and digital parts. Each object will become a member of the network and communicate with other objects to achieve a common objective.

3.2 Types of Data Preprocessing Technique

Three different kinds of preprocessing methods exist.

- Rescale Data
- Binarize Data
- Data Augmentation

3.2.1 Rescale Data

ML algorithms gain most from rescaling characteristics when our data has multiple-sized properties. To maintain the uniformity of the dataset's measurement parameter, all dataset attributes must have the same scale. This is also used in an optimization procedure to maintain the consistency of the data collection.

3.2.2 Binarize Data

Any entity's data attributes can be converted into binary integers using the "binarization" technique. It is done in order to classify algorithms more successfully. Data can be converted into binary using a binary threshold. Any value that is over the threshold is recorded as 1, and any value that is at or below the threshold is recorded as 0. In this instance, your data is being binaryized. When you want to give Crip value because you have value, it can be useful.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

Article Title: IoT Network Traffic Classification using Deep Learning Algorithms

3.2.3 Data Augmentation

Using a technique known as "data augmentation," experts or researchers can increase the amount of data available for model training even in the absence of data collection or acquisition. It solely pertains to increasing the volume of data by using the insights gleaned from training data. We might need more data with as many variations as is practical in order to get better generalizations. However, the dataset is too small to capture variance in a meaningful way. In these situations, data augmentation is quite useful and essential.

3.3 Feature Selection Chi-Square

Feature selection, sometimes called attribute selection, is the process of choosing several appropriate characteristics among the dataset and then utilizing machine learning techniques to enhance the model's performance. An enormous number of unnecessary features exponentially increases training time and increases overfitting risk. To assess categorical features in a dataset, apply the chi-square approach. Determine the necessary quantity of features with the highest Chi-square values by calculating the Chi-square between every feature and the goal. It evaluates if there is a true relationship between two categorical variables in the population and the relationship that exists between them in the sample. The supervised feature selection approach can be used to select input characteristics pertaining to deemed being the greatest beneficial creating an illustration. We use either wrapper techniques or filter-based approaches for our supervised feature selection strategy. The best features are chosen using a wrapper-based technique, such as Recursive Feature Elimination (RFE). However, we used the filter-based feature selection strategy, which scores the connection among the goal labels, or class labels, and the features, to choose the features from our feature space F. Because the goal class labels are categorical in nature and some input attributes are analytical, or upon preliminary processing, they change to quantified, we specifically chose the ANOVA (Analysis of Variance) F-value feature selection approach.

3.4 Convolutional Neural Networks (CNN)

At the moment, CNN is the go-to source for anything pertaining to images. They at the moment, CNN is the go-to source for anything pertaining to images. In terms of accuracy, they perform better than their rivals. It works well in other domains as well, like recommender systems and natural language processing. CNN's primary advantage over its predecessors is its use of machine learning to autonomously identify important components. With multiple photos of dogs and cats as an example, it can independently determine the distinct traits of each class. CNN is also computationally efficient. It carries out pooling, some convolution, and parameter sharing. Because of this, CNN models are compatible with all devices and have a global appeal.

Everything about this sounds like pure delight. Utilizing a remarkably strong and efficient model that autonomously extracts characteristics for unparalleled precision.

3.5 CNN Classification

- **Input Layer**: The input image data enters this layer.
- **Convolutional Layer:** This layer takes the input image and runs it through a series of filters to extract low-level features like corners and edges. The rectified linear unit (ReLU) activation function receives the output of the convolutional layer. Other activation functions are also available for use. This causes the model to become non-linear. Learning the spatial and temporal correlations between the pixels in an image is an area in which CNNs excel. They achieve this by applying a number of filters to the input image through a sequence of convolutional layers. Convolutional layers extract significant elements from the input image, including edges, lines, textures, and forms, by applying a collection of filters to the image. A feature map is generated by convolving each filter, which is a tiny matrix of weights, with the input image. What the CNN has learned to identify in the input image is represented by each feature map, which is a specific feature or pattern. Every filter's convolution operation produces an output that may be understood as the feature maps. Several feature maps, each representing a distinct set of learnt characteristics, are produced by the convolutional layer through the application of extra filters. To add non-linearity to the training process, the output of the convolution layer is passed to an activation layer (like ReLU). Next, the pooling layer receives the output from the activation layer.
- **Pooling Layer:** The feature maps created by the first convolutional layer are down-sampled by this layer, which lowers their dimensionality and improves processing efficiency. In addition to the convolutional layers, CNNs frequently include pooling layers that down sample the feature maps produced by the convolutional layer. Consequently, the dimensionality of the feature maps decreases, improving their processing effectiveness and reducing the likelihood of overfitting. The max-pooling process uses the maximum value of each non-overlapping region that makes up the feature maps. This method creates a compressed version of the input image, retaining the salient elements of the feature maps and removing the remainder.
- **Max Pooling Layer:** The unique characteristics of mappings produced during the process second convolutional layer are further reduced in dimensionality by this layer, which down-samples them.
- **Fully connected or dense layers:** After being traversed all of them completely connected layers (also called dense layers) with the output from the convolutional and pooling layers, the input image is classified, or other tasks, including segmentation, are

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

carried out using a typical neural network design. A collection of high-level features that accurately capture the salient aspects of the input image are the result of the convolutional and pooling layers of a CNN. Using these acquired features, the fully connected layers' job is to predict the class or label of the incoming image.

3.6 Segmentation

Segmenting image pixels into classes that are semantically interpretable as actual objects allows for semantic classification of the pixels. The technique of employing CNN to classify annotation and region proposal refer to the division of pixel values into discrete groupings. Additionally, "region proposals" can also refer to small groups of pixels that are probably related to the same object. CNNs for semantic segmentation often use a fully convolutional network architecture, which replaces the totally connected convolutional structure of a typical neural network. With an identity assigned to each pixel, the system is able to process input images of any quality and generate a final pattern of the same measurement.

3.7 Model Evaluation

Through this evaluation approach, we can ascertain which algorithm, given the dataset, will be able to tackle a specific problem satisfactorily. Using the same input dataset, it evaluates the performance of many CNN models. The evaluation method heavily emphasizes the prediction ability of the model. From all the algorithms we use in this stage, we select the one that yields the best accuracy for the input data and is considered the best model since it produces a more accurate outcome forecast. Accuracy is considered to be crucial when using CNN to tackle a range of problems. There are several ways to tackle a CNN problem: gathering data, defining the problem, coming up with ideas using the data at hand, preprocessing, transformation, training the model, and evaluation. Although a CNN model has several parts, the evaluation phase is the most crucial since it lets us determine how well the model foretells the future. The CNN model's application and performance are evaluated using the final accuracy measurements.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

Article Title: **IoT Network Traffic Classification using Deep Learning Algorithms**

4 Result and Discussion

	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 2: *Input Dataset*

Figure 2 shows the model of the dataset. To ascertain whether the byte sequences obtained using the proposed method provide valuable information for classifying IoT networks, the results from many dataset samples.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

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 3: *Null Values*

Figure 3 illustrates the preprocessing techniques. The purpose of this method is to locate null and repeating values inside 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 4: *Attacks*

Figure 4 demonstrate the attacks in network, which includes, Normal, Generic, Exploits, Fuzzers, Dos, Reconnaissance etc.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

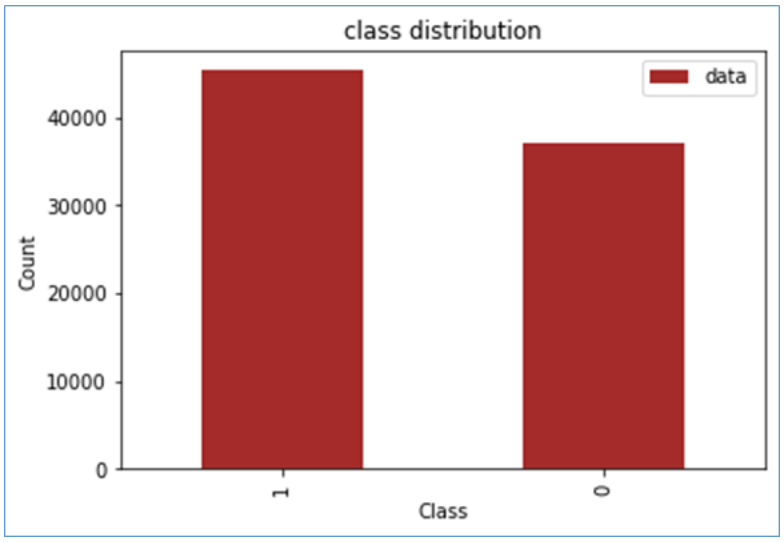

Figure 5: *Class Distribution*

The distributions of the class model are displayed in Figure 5 a dictionary with the amount of randomly generated samples to include in the dataset as the value and the class value (e.g., 0 or 1) as the key represents a class distribution.

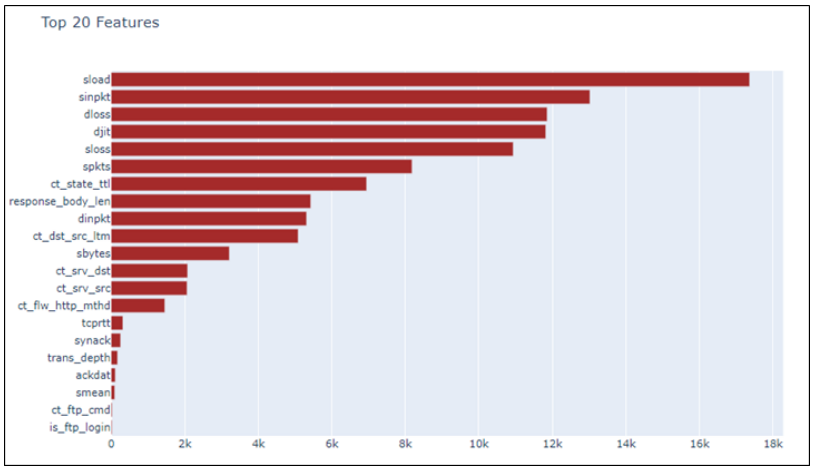

Figure 6: *Top Features*

The process of selecting a unique visual image with a chi-square is shown in Figure 6 One statistical method for comparing two or more statistical data sets is the chi-square computation. It permits the assessment of dichotomous independent variables as well as multiple group investigations.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

Article Title: IoT Network Traffic Classification using Deep Learning Algorithms

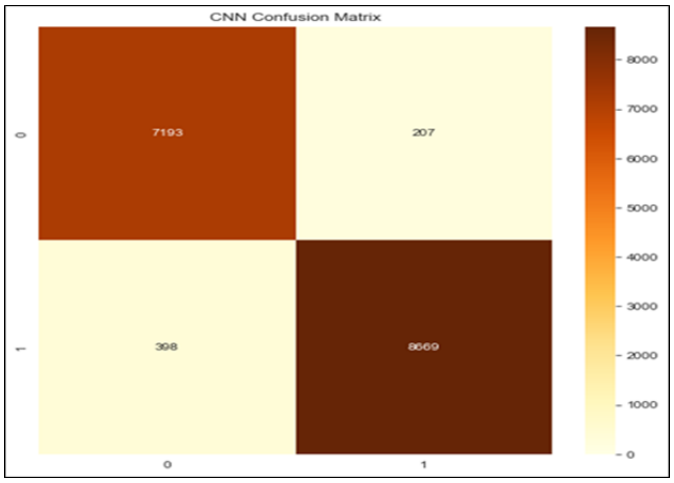

Figure 7: *Convolutional Neural Network Classifier*

The CNN classifier is shown in Figure 7 These layers are traversed by an image classifier in order to produce a classification. Using filters to scan the image, the convolutional layer pulls features from the image.

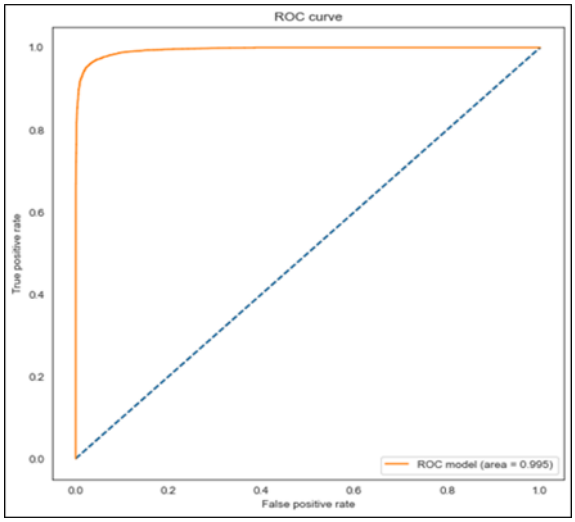

Figure 8: *ROC Curve*

The CNN classifier is shown in Figure 8. These layers are traversed by an image classifier in order to produce a classification. Using filters to scan the image, the convolutional layer pulls features from the image.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03(03), MARCH 2024, pp. 23-37

Article Title: IoT Network Traffic Classification using Deep Learning Algorithms

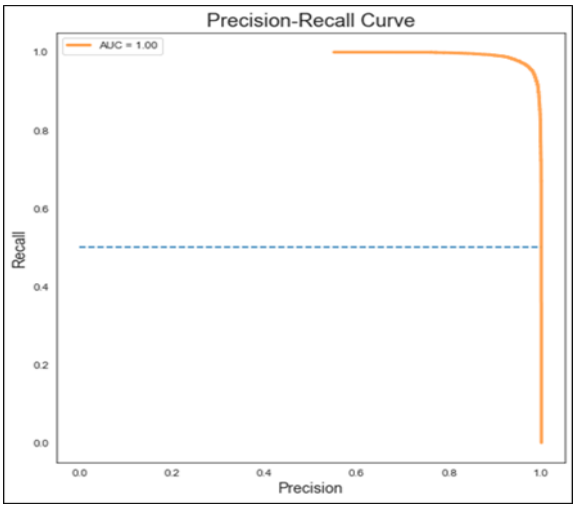

Figure 9: *Precision-Recall Curve*

This recall curve is shown in Figure 9 Recall is defined as the proportion of accurate true positive classifications (true positives) to the total number of true positive classifications (true positives plus false negatives). Each graph with recall values on the x- and precision values on the y-axes is known as a PR curve. Predicting the probability of an observation falling into each class, rather than the actual classifications, may be more flexible in a classification problem.

5 Conclusion

This paper shows how to use deep learning techniques for traffic analysis to identify IoT devices and the network flows they generate. To exclude null and repetitive values from the input dataset, apply the data preparation approach. The feature selection process employs the chi-square approach. Processing the extraction areas using the Chi-Square method allows for the extraction of many features and the selection of key elements for enhanced classification. The data classification stage comes after the extraction stage. Convolutional neural networks (CNN) are used for classification. CNN classification forecasted the model's performance analysis. In a real-world IoT scenario, this project provides a useful framework for categorizing IoT network traffic.

References

1. Hamid Tahaei; Firdaus Afifi; Adeleh Asemi; Faiz Zaki; Nor Badrul Anuar, Year: 2020, “The rise of traffic classification in IoT networks: A survey”, Journal of Network and Computer Applications, Vol: 154, pp. 102538.

2. Dana Haj Hussein; Mohamed Ibnkahla, Year: 2022, "An IoT traffic modeling framework and its application to autonomous edge scaling", IEEE Global Communications Conference (GLOBECOM), Rio de Janeiro (Brazil), pp. 5656 – 5661.

3. Geethapriya Thamilarasu; Adedayo Odesile; Andrew Hoang, Year: 2020, "An intrusion detection system for internet of medical things", IEEE Access, Vol: 8, pp. 181560 – 181576.

4. Manabhanjan Pradhan; Sujata Mohanty; A. Ocean Seemona, Year: 2022, "Machine learning-based intrusion detection system for the internet of vehicles", In 5th International Conference on Computational Intelligence and Networks (CINE), Bhubaneswar (India), pp. 1 – 6.

5. Mahmoud Abbasi; Amin Shahraki; Amir Taherkordi, Year: 2021, "Deep learning for network traffic monitoring and analysis (ntma): A survey", Computer Communications.

6. Ivan Cvitić; Dragan Peraković; Marko Periša; Mate Botica, Year: 2021, "Novel approach for detection of IoT generated DDoS traffic", Wireless Networks, Vol: 27, no: 3, pp. 1573 – 1586.

7. Shuren Li; Yifei Lu; Jingqi Li, Year: 2022, "CAD-IDS: A cooperative adaptive distributed intrusion detection system with fog computing", In IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hangzhou (China), pp. 635 – 640.

8. Ali Bou Nassif; Manar Abu Talib; Qassim Nasir; Fatima Mohamad Dakalbab, Year: 2021, "Machine learning for anomaly detection: A systematic review", IEEE Access, Vol: 9, pp. 78658 – 78700.

9. Ridi Ferdiana; Year: 2020, "A systematic literature review of intrusion detection system for network security: Research trends, datasets and methods", In Proceedings of the 4th International Conference on Informatics and Computational Sciences (ICICoS), Semarang (Indonesia), pp. 1 – 6.

10. Eman Ashraf; Nihal FF Areed; Hanaa Salem; Ehab H. Abdelhady; Ahmed Farouk, Year: 2022, "IoT based intrusion detection systems from the perspective of machine and deep learning: A survey and comparative study", Delta University Scientific Journal, Vol: 5, no: 2, pp. 367 – 386.

11. Rajesh Kalakoti; Sven Nõmm; Hayretdin Bahsi, Year: 2022, "In-Depth Feature Selection for the Statistical Machine Learning-Based Botnet Detection in IoT Networks", in IEEE Access, Vol: 10, pp. 94518 – 94535.

12. Houda Jmila; Gregory Blanc; Mustafizur R. Shahid; Marwan Lazrag, Year: 2022, "A Survey of Smart Home IoT Device Classification Using Machine Learning-Based Network Traffic Analysis", in IEEE Access, Vol: 10, pp. 97117 – 97141.

13. Amir Mahdi Sadeghzadeh; Saeed Shiravi; Rasool Jalili, Year: 2021, "Adversarial Network Traffic: Towards Evaluating the Robustness of Deep-Learning-Based Network Traffic Classification", in IEEE Transactions on Network and Service Management, Vol: 18, no: 2, pp. 1962 – 1976.

14. Elaiyasuriyan Ganesan; I-Shyan Hwang; Andrew Tanny Liem; Mohammad Syuhaimi Ab-Rahman, Year: 2021, "SDN-enabled FiWi-IoT smart environment network traffic classification using supervised ML models", In Photonics, Vol: 8, No: 6, p. 201, Multidisciplinary Digital Publishing Institute.

15. M. Meenaxi Raikar; S. M. Meena; Mohammed Moin Mulla; Nagashree S. Shetti; Meghana Karanandi, Year: 2020, "Data traffic classification in software defined networks (SDN) using supervised-learning", Procedia Computer Science, Vol: 171, pp.2750 – 2759.