Spring 2023

Classification of Darknet Traffic by Application Type

Shruti Sharma
San Jose State University

Classification of Darknet Traffic by Application Type

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Shruti Sharma

November 2022

The Designated Project Committee Approves the Project Titled

Classification of Darknet Traffic by Application Type

by

Shruti Sharma

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

November 2022

Dr. Mark Stamp	Department of Computer Science
Dr. William Andreopoulos	Department of Computer Science
Samanvitha Basole	Software Engineer

ABSTRACT

Classification of Darknet Traffic by Application Type

by Shruti Sharma

The darknet is frequently exploited for illegal purposes and activities, which makes darknet traffic detection an important security topic. Previous research has focused on various classification techniques for darknet traffic using machine learning and deep learning. We extend previous work by considering the effectiveness of a wide range of machine learning and deep learning technique for the classification of darknet traffic by application type. We consider the CICDarknet2020 dataset, which has been used in many previous studies, thus enabling a direct comparison of our results to previous work. We find that XGBoost performs the best among the classifiers that we have tested.

ACKNOWLEDGMENTS

I want to thank Dr. Mark Stamp, Samanvitha Basole and Dr. William Andreopoulos for their constant support and guidance over the course of entire CS-298 project.

TABLE OF CONTENTS

CHAPTER

CHAPTER 1

Introduction

Over the last few decades, we have seen an increased use of the Internet which is also commonly known as the World Wide Web, however, only a handful of us may have heard about the darkweb. Darkweb is also a Worldwide Network but its content only exists in Darknet (a separate network) and requires unique access permissions. Darknet helps in interaction with the network anonymously without revealing any relevant information like location or identity [1]. Darkweb accounts for 48% of the Internet while the surface web (which we can normally access) accounts for only 4% of the Internet [2]. According to [3] there are different layers of the Internet and to pass data from one layer to another layer, relays of Tor are used.

The darknet is used for a variety of illicit and unapproved activities, including drug use, terrorism, child pornography, and human trafficking. Also, [3] have stated the importance of traffic analysis on the dark web to detect any malicious intent of attackers who are continuously changing their techniques to avoid detection of security risk. Furthermore, [4] has shown the traffic generated by different application categories on Darknet.

Figure 1: Layers of Internet [3]

As stated by [3], traffic classification is important to implement security policies

and is considered a crucial step in network management. Hence our research further aims to improve Darknet classification using Machine Learning methods to classify different traffic sources and the application category of darkweb. The research is inspired by work proposed in [4] and builds upon it to experiment with different machine learning technologies that could further improve the traffic classification of darkweb. We propose to test the impact of Gradient Boosting Classifier, XGBoost, K-Nearest Neighbors (KNN), Multi-layer Perceptron (MLP), and Residual Network(ResNet) on traffic and application classification for different SMOTE levels.

Let us discuss the flow of context in this paper. The background of Tor and Virtual Private Networks (VPN) is highlighted in Chapter 2, which also covers related research in the area of traffic classification. The specifics of the dataset, the preprocessing procedures, and the methods used to test for classification are covered in Chapter 3. The findings of our experiments and the investigation of potential future work are covered in Chapter 4. Finally Chapter 5 represents the possible implementations that could be used in the future to improve the performance.

CHAPTER 2

Background

Here we will talk about the categories of datasets, and some related work for the topic we chose. This section mainly explains the background and previous work that has been researched and techniques applied to resolve the problem.

2.1 Tor

Tor provides anonymity to your IP address by using special encryption techniques and hides browsing activity by redirecting the web traffic through various node [5]. Tor also known as the onion router was first initialized by the US navy for encrypted communication and now has become an open source and nonprofit with more than thousands of routers and used by millions of users [6]. Due to the anonymous use of Tor it is mostly used for unethical purposes by hackers, activists and cybercriminals for selling weapons, drugs etc [7], and hence there is a need for a systematic technique to classify users of Tor. The anonymity of Tor attributes to it being a distributed network of TCP-based applications where a user enters a path in the circuit and each node in the network just knows a node before and after it [8].

The architecture of the onion router Figure 2 describes how the router works. As the data moves through the network, decryption is done at every node and is sent to the next router or node [9].

2.2 Virtual Private Networks

A VPN is like a filter that provides anonymity to IP addresses as it runs through a host server which becomes the source of the data for the user search. This provides security and hides information about the user from the service provider [10]. The Internet can be easily accessed from anywhere using any available access media, including analog modems, ISDN, cable modems, DSL, and wireless, thanks to VPN architecture's dependable authentication system. Local area network (LAN) VPN

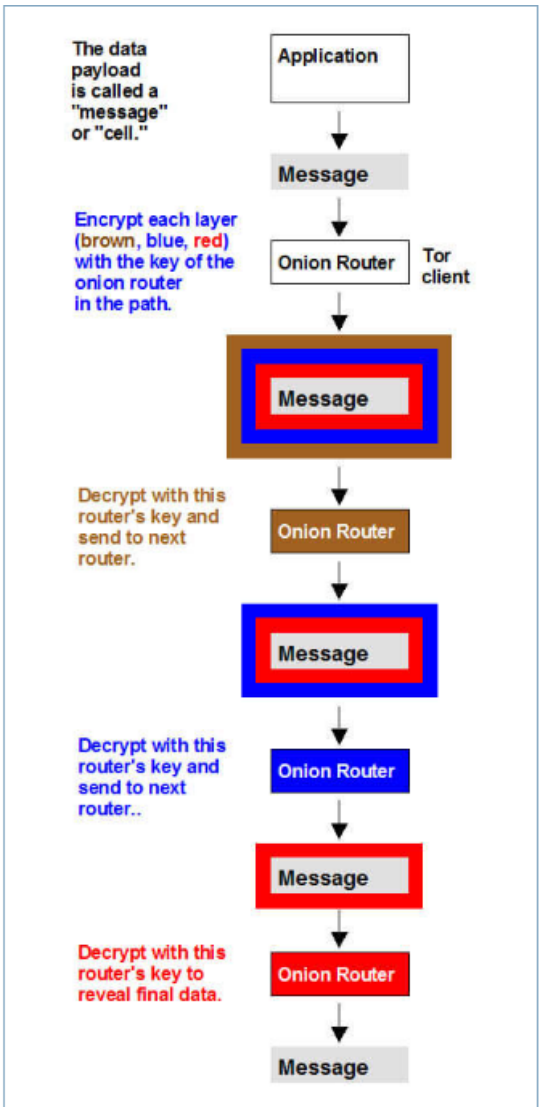

Figure 2: Onion router architecture [9]

services, dial-up VPN services, and Ethernet VPN services are the three main categories of VPN services [11]. Figure 3 shows the secured connection between the client and the server. This also helps in accessing many blocked sites while preserving the hidden IP address [12].

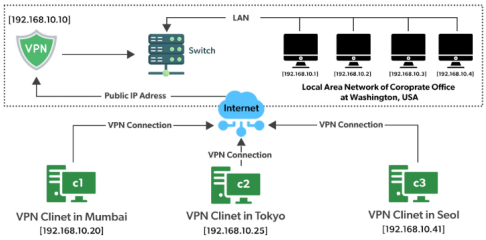

Figure 3: VPN architecture [12]

2.3 Related work

Some studies on darknet traffic were conducted, although the dataset available in those studies were limited. The classification of dark entities was studied utilizing feature importance and machine learning methods like ROC analysis in [13]. They employed two classification tasks: one was the multiclass classification of Tor, Non Tor, VPN, and Non VPN, and the other was the categorization of darknet and benign data. Their final results showed that Random Forest performed both tasks with an accuracy of 98%. CICDarknet2020 was the dataset in use at the time.

On the other hand, [14] has attempted to separate Tor traffic from non-Tor traffic and has gone further to categorize the Tor traffic into various applications like browsing, chat, file transfer, etc. using time constraints. They created 2 profiles that make use of sites like Facebook and Skype in order to produce a representative dataset of actual traffic. The dataset included eight distinct traffic types from 18 applications. They used feature selection and tested on algorithms including Zero Rule, K-Nearest Neighbors, and C4.5 sing 10 folds cross-validation as part of their evaluation methodologies. According to their findings, C4.5 worked best when precision and recall turned out to be greater than 0.9. It was ultimately determined that time-based approaches are more effective at differentiating between Tor and non-Tor traffic as well as across various applications classification.

Comparatively, [7] has used the UNB-CIC Tor network dataset to explore the distinction between Tor and Non-Tor Traffic in more detail. The trials were carried out using deep learning algorithms [7]. After conducting their tests, they found that deep neural networks classified Tor and non-Tor traffic with an accuracy of 99.89%. Moreover, adversarial samples produced by a Generative Adversarial Network were used to assess how robust the suggested DNN classifier was. It was found that the DNN classifiers failed to identify all of the hostile cases. The DNN resilience against the adversarial attack was ultimately increased by additional retraining with adversarial samples.

Additionally, [15] has suggested a hierarchical categorization method for darknet traffic that can distinguish between 25 different user behaviors as well as four different types of darknet clients, including Tor, I2P, ZeroNet, and Freenet. 26 time-based features were retrieved from the dataset, which was created using actual darknet traffic from all varieties of darknet. The results showed that user behavior could be predicted with 91.6% accuracy while the hierarchical classifier distinguished the customers with 96.9% accuracy.

In a different study, [16] extended the work of [13] by applying SMOTE methods for balancing the traffic data and by appropriately selecting features by reducing the number of features from 63 to 8 for benign and darknet traffic, 8 for multiclass Tor, Non Tor, VPN, and Non VPN, and 6 for subtraffic classes from the original 63. It was determined that the SMOTE techniques and Random Forest model produced the maximum classification accuracy, with 97.22% accuracy for benign and darknet traffic, 97.16% accuracy for the four main types of traffic, and 85.99% accuracy for subclasses of traffic.

The CICDarknet2020 dataset was utilized in one study by [17], which also conducted feature importance analysis using the chi-squared test for feature selection.

Utilizing conditional generative adversarial networks for oversampling, the imbalance was handled, and traffic was classified using a random forest classifier, yielding an F1-score of 97.8

Our research has been inspired by the work of [4]. It is built over the groundwork in [4] that describes the traffic classification adversarial attacks of the darknet using various machine learning techniques like SVM, Random Forest (RF), CNN, and ACGAN. The dataset used by the team was also CICDarknet2020. The results of different classification algorithms in Table 1 and Table 2 found Random Forest (all SMOTE levels) and CNN (no SMOTE) with the best F1-score. For application classification, random forest alone produced the best results. The boxed numbers below in Table 1 and Table 2 show 99.8% F1-score for traffic classification and 92.2% F1-score for application classification. Taking further the work of [4] we test new algorithms like KNN, GBC, XGBoost, MLP and ResNet to attempt a further improvement of the F1-score and accuracy.

Table 1: Traffic classification F1-score at various SMOTE levels [4]

Learning technique	SMOTE percentage 0%	20%	40%	60%	80%	100%
SVM	0.986	0.993	0.993	0.993	0.993	0.993
RF	0.998	0.998	0.998	0.998	0.998	0.998
CNN	0.998	0.995	0.995	0.995	0.996	0.995
AC-GAN	0.974	0.980	0.984	0.986	0.987	0.987

Table 2: Application classification F1-score at various SMOTE levels [4]

Learning technique	SMOTE percentage 0%	20%	40%	60%	80%	100%
SVM	0.834	0.839	0.842	0.846	0.847	0.848
RF	0.922	0.920	0.921	0.921	0.920	0.920
CNN	0.887	0.883	0.883	0.887	0.888	0.885
AC-GAN	0.738	0.750	0.762	0.768	0.767	0.759

CHAPTER 3

Implementation

This chapter explains the implementation details for the project and evaluates the results we obtain. Its starts with describing the data set, the pre-processing steps involved and different model implementations along with their results.

3.1 Dataset

The data set we used for traffic classification is the CICDarknet2020 data set published by the Canadian Institute for cybersecurity. The data set is the combination of two public datasets named ISCXTor2016 and ISCXVPN2016. The data set consists of 24,311 dark net samples while 134,348 benign samples. The first layer of traffic comprises of Tor, Non-Tor, VPN, and Non-VPN. The darknet samples have been further divided into different application categories like VOIP, video streaming, P2P, audio- streaming, browsing, chat, email, and file transfer as in Table 3 and Figure 4 respectively.

Table 3: Samples per application category [18]

Class	Application Type	Samples
0	Audio-Streaming	18,065
1	Browsing	32,809
2	Chat	11,479
3	Email	6,146
4	File Transfer	11,183
5	P2P	48,521
6	Video-Streaming	9,768
7	VOIP	3,567

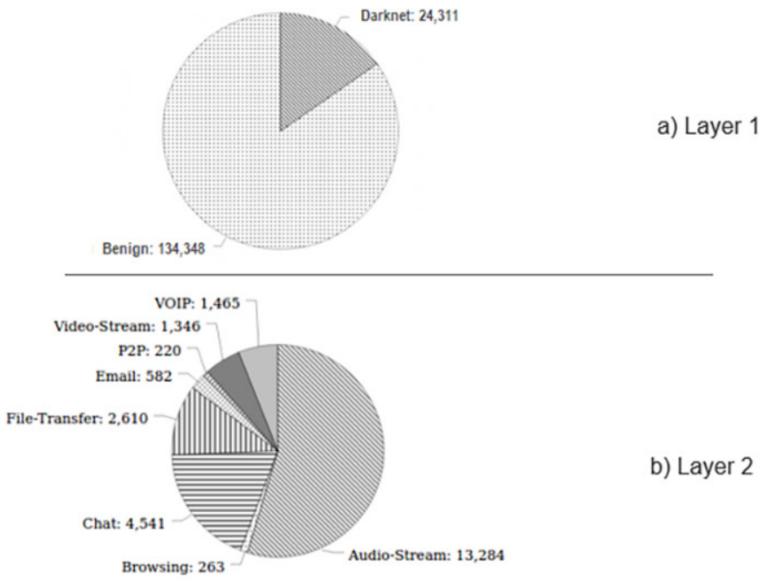

Figure 4: Dataset details [18]

3.2 Data Processing

For our model, we have taken numerous data pre-processing steps. Our dataset consists of some null sample values which have been dropped from the dataset as they do not significantly contribute to the classification accuracy. We have dropped irrelevant features like **FlowID** and **Timestamp** while we have created a unigram, bigram, trigram octet for **Source** and **Destination IP**. Feature encoding and min-max normalization were used on numerical features. Finally, we were left with 72 columns after the preprocessing step as in Appendix A.

Our dataset with 72 features did not appear to be a balanced dataset for both traffic and application classification. Hence we used oversampling of minority classes to study the effect of unbalanced data on the accuracy. SMOTE method was used at 0%, 20%, 40%, 60%, 80%, and 100% levels to balance the dataset for different traffic classes and application classes.

3.3 K-Nearest Neighbors

KNN is a classification technique that helps in classifying labels as per its k nearest neighbors. In KNN K is the hyperparameter. There are various distance metrics that are used by KNN to measure the distance, some of them being minkowski distance, manhattan distance, euclidean distance, cosine distance, hamming distance, and jaccard distance [19]. If k is too small, there is a high chance of overfitting the model while a bigger k would result in a simpler model. We have used the k value of 5 in our paper for different SMOTE levels. Figures 5 and 6 is the sample confusion matrix for different SMOTE levels of classification.

Figure 5: Application classification for 60% SMOTE level

3.4 Gradient Boosting Classifier

We have used Gradient Boosting Classifier on our dataset as it has continuously proven to be one of the strongest techniques in classification prediction. Gradient Boosting classifier combine multiple weak learners to make them into one strong learning algorithm. Here decision trees or regression trees are used as weak learners. Since there is a chance that gradient boosting may have the problem of overfitting

Figure 6: Application classification for 100% SMOTE level

the training data we need to use different regularization methods like L1 and L2 regularization to balance the weight. For our experiments, we have used the loss function as log loss, learning rate as 0.1, and n_estimators as 100. Our sample confusion matrices are mentioned in Figure 7 and Figure 8 for different SMOTE levels.

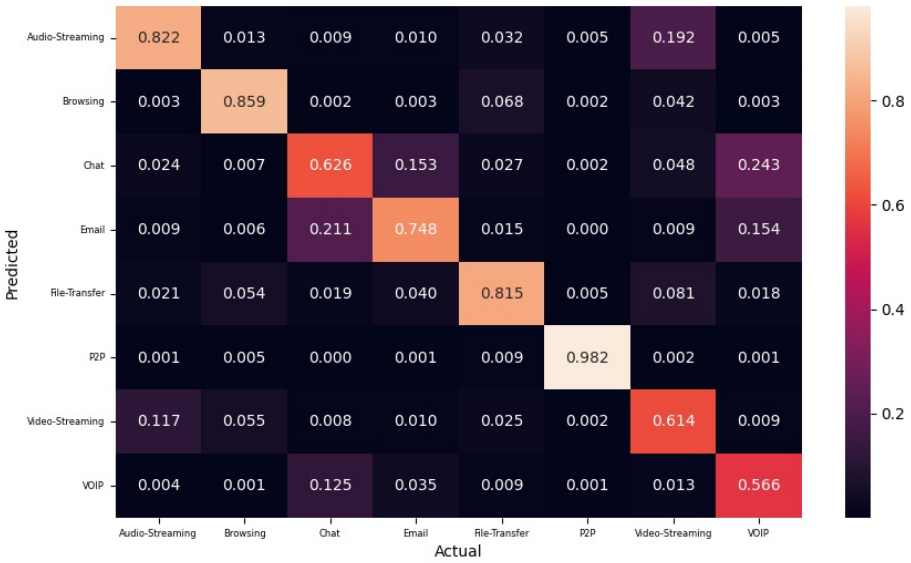

Figure 7: Application classification for 0% SMOTE level

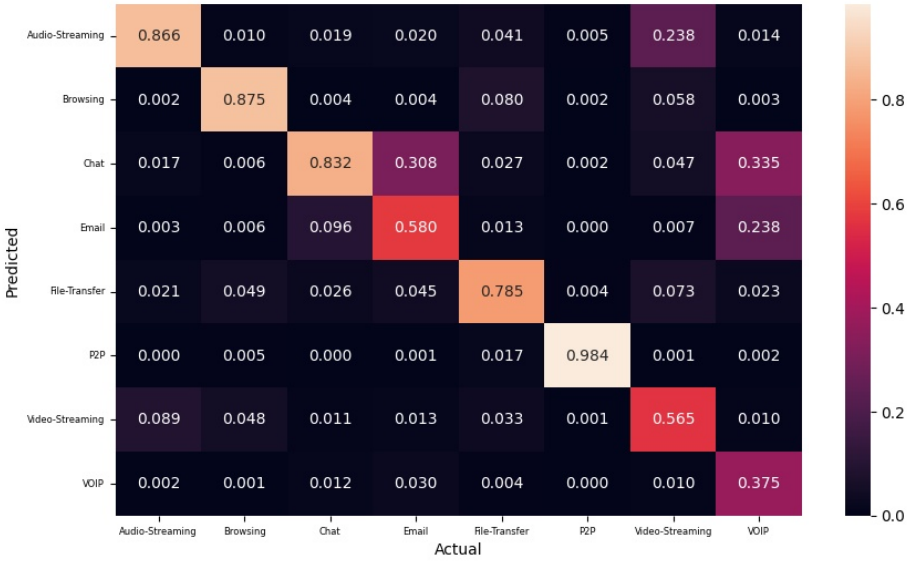

Figure 8: Application classification for 40% SMOTE level

3.5 XGBoost

XGBoost is an ensemble learning technique but with a refined version of gradient boosting classifier hence known as extreme gradient boosting. It provides boosting of trees in parallel. It is almost similar to gradient boosting but with the advantage of better performance and better regularisation. In our experiments we have used gbtree as a booster, a verbosity of 1, learning rate of 0.3, a depth of 6, and a uniform sampling method. Figure 9 and 10 show the results for different SMOTE levels.

3.6 Multilayer Perceptron

Just like neural networks, MLP is a fully connected artificial neural network consisting of an input layer, hidden layer, and output layer. MLP provides solutions for extremely complex problems. MLP takes vectors as input. In our problem statement, we are using MLP with 100 hidden layers and Rectified linear unit as the activation function. In this paper 100 hidden layers of the neural network are used with the ReLU activation alongside ADAM solver and an alpha value of 0.0001. The confusion matrix is shown in Figure 11 and Figure 12 for different SMOTE levels.

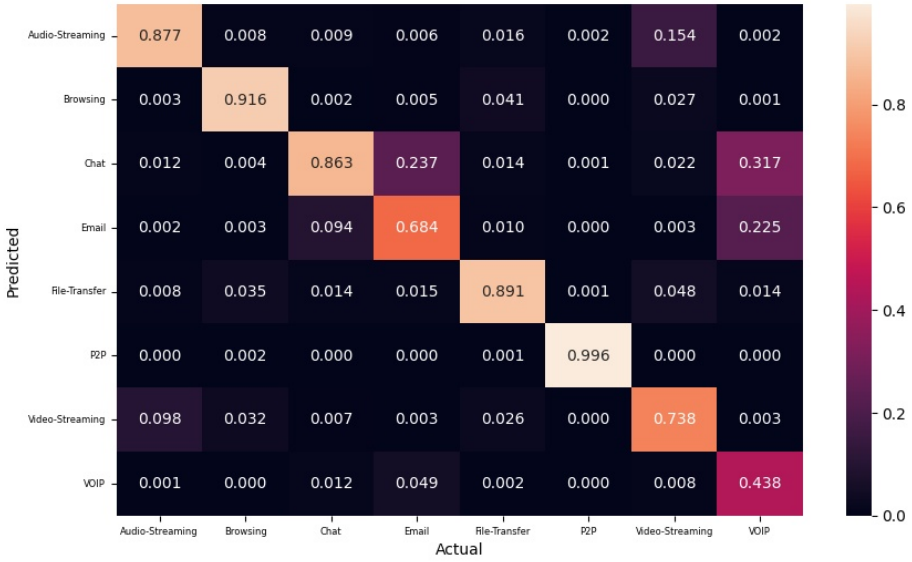

Figure 9: Application classification for 20% SMOTE level

Figure 10: Application classification for 80% SMOTE level

3.7 ResNet

ResNet is an artificial neural network that takes images as input. It helps resolve the problem of vanishing and exploding gradient descent. They use skip connections (as shown in Figure 13) in deeper networks to solve the problem and these are also

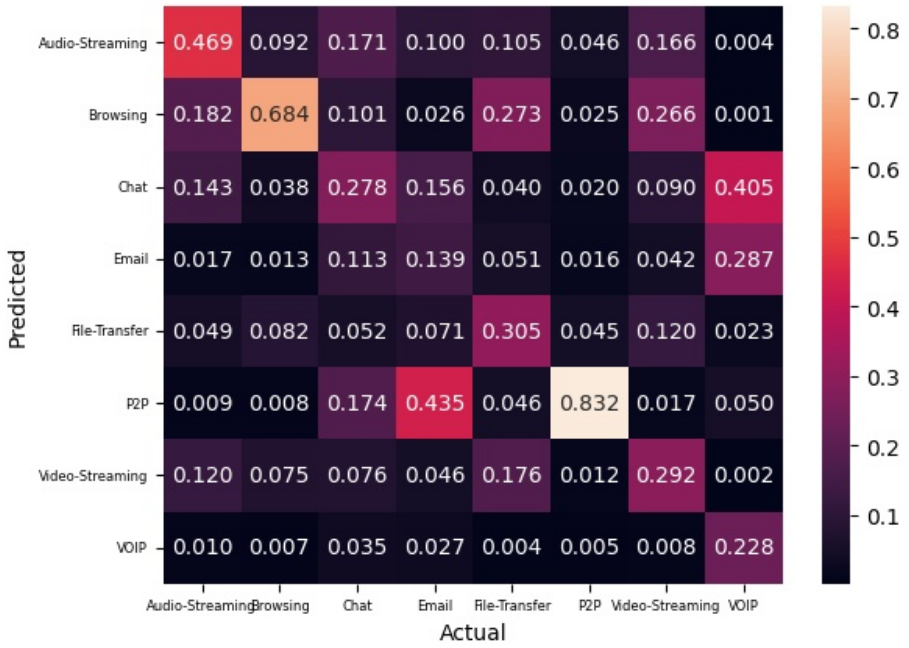

Figure 11: Application classification for 80% SMOTE level

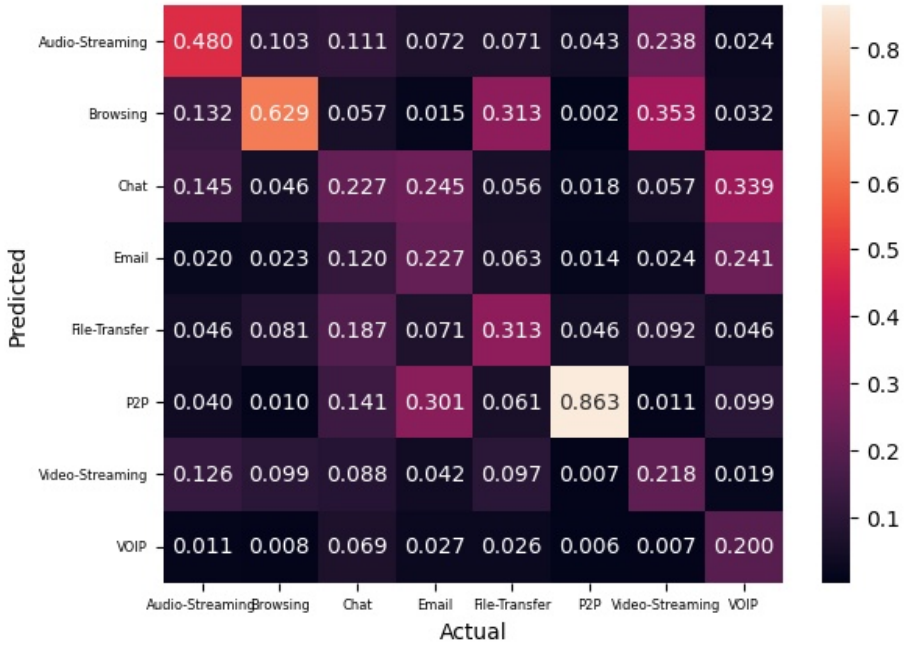

Figure 12: Application classification for 100% SMOTE level

known as residual blocks. The purpose of a skip connection is to connect the activation layer to another layer by skipping some of the layers between them. We have trained ResNet on 32 epochs on images of 9×9, 9×9 with random forest feature weights, and 9×9 random forest spiral to get the results for the shape with the highest accuracy. The confusion matrix for SMOTE levels 40% and 80% with 9×9 random forest spiral feature weights are mentioned in the Figure 14 and Figure 15.

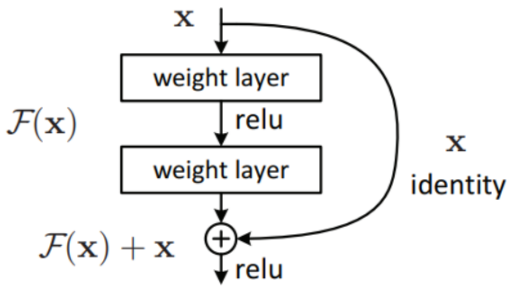

Figure 13: ResNet architecture [20]

Figure 14: Application classification (9×9 with RF and spiral) for 40% SMOTE level

Figure 15: Application classification (9×9 with RF and Spiral) for 80% SMOTE level

3.8 Evaluation Metrics

One approach to gauge how frequently a machine learning classification algorithm classifies a data point properly is to look at its accuracy. For our experiments, we have used accuracy and F1-score for the individual performance of different classifiers. Accuracy is simply the ratio of correct classifications to the total number of classifications, while the F1-score is computed as the harmonic mean of the precision and recall

$$\texttt{F1} = 2 \times \frac{(\texttt{Precision} \times \texttt{Recall})}{(\texttt{Precision} + \texttt{Recall})}$$

where

$$\texttt{Precision} = \frac{\texttt{True Positives}}{(\texttt{True Positives} + \texttt{False Positives})}$$

and

$$\texttt{Recall} = \frac{\texttt{True Positives}}{(\texttt{True Positives} + \texttt{False Negatives})}$$

CHAPTER 4

Results

We will talk about a variety of experimental methods in this part. We will start by identifying the 2-D picture representation strategies that we employed for the ResNet studies. The SMOTE trials, which were utilized to address the data imbalance issue, are also taken into account.

4.1 Image Representation of Data

We created 9×9 images using our dataset, where each pixel stands for a feature, and the remaining pixels are produced by padding. Figure 16 represents samples from the application class with pixels ordered by random forest weights. The Darknet dataset was employed with the premise that since convolutions act on local structures, doing so will improve the performance of the classifier. The pixels are arranged in the dataset according to the order of feature representation.

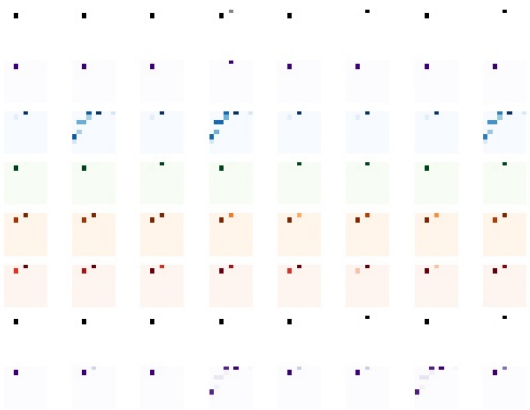

Figure 16: Application data as 2-D images ordered with RF feature weights

The results from Table 4 show that the Resnet performed better with original 9×9 compared to random forest feature weights. In contrast, for the traffic classification dataset as in Table 5 the shape with maximum performance was the shape with

random forest feature importance, followed by the shape with random forest feature importance and centered.

Table 4: Performance on application classification for 2-D representations of data

Data Representation	Accuracy	F1-score
Original with 9×9 shaped	0.814	0.806
9×9 shaped with RF feature importance	0.730	0.741
9×9 shaped with RF feature importance and centred	0.785	0.790

Table 5: Performance on traffic classification for 2-D representations of data

Data Representation	Accuracy	F1-score
Original with 9×9 shaped	0.955	0.955
9×9 shaped with RF feature importance	0.965	0.965
9×9 shaped with RF feature importance and centred	0.962	0.962

4.2 ResNet Experiments

In our Resnet, we have applied feature vectors from 9×9, 9×9 with random forest feature weights, and 9×9 random forest centered with 32 epochs. Model accuracy and model loss were computed for all the shapes with different SMOTE levels. From Figure 17 that represents 0% SMOTE level for the original shape, it is observed that though there is some unevenness in the test accuracy as the number of epochs changes there still exists some significant increase. Also, as shown in Figure 18 the test loss does not seem to converge significantly. On the other hand with 9×9 random forest feature weights as in Figure 19 that has 80% SMOTE level the accuracy increases and finally converges after 25 epochs,with a converging loss graph as shown in Figure 20 ranging for loss around 12% after 30 epochs.

4.3 SMOTE Experiments

We have computed the results for different algorithms in our classification. Different SMOTE levels were used for each classification algorithm.

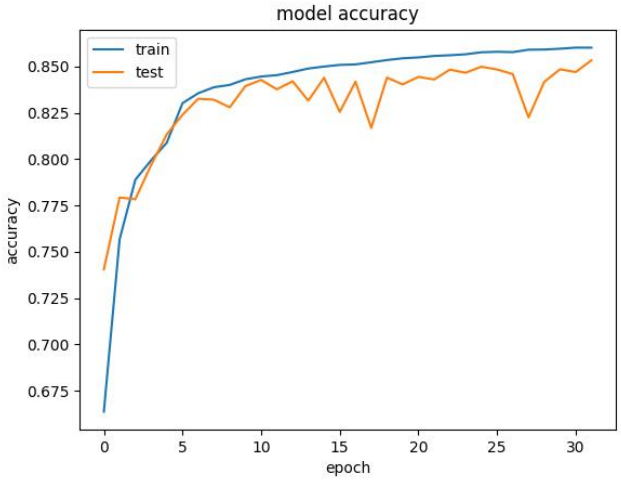

Figure 17: Training vs test accuracy for application data in original shape

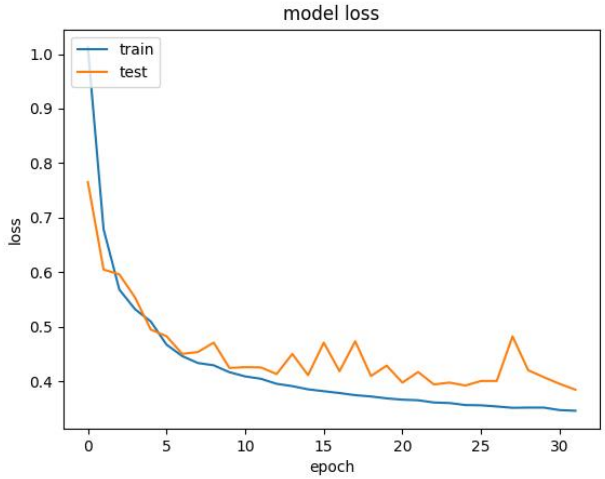

Figure 18: Training vs test loss for application data in original shape

Our research revealed that the performance of the KNN, Gradient Boosting, XGBoost, MLP, and ResNet algorithms were not significantly improved by increasing the SMOTE level. Our results demonstrate that when no SMOTE was applied, XGBoost performed better for traffic classification, with a maximum accuracy of 0.983 as seen in Table 6 and a matching F1-score of 0.983 in Table 7. Also, with an F1-score of 0.893 in Table 8 and 0.896 accuracy in Table 9 XGBoost took a lead for

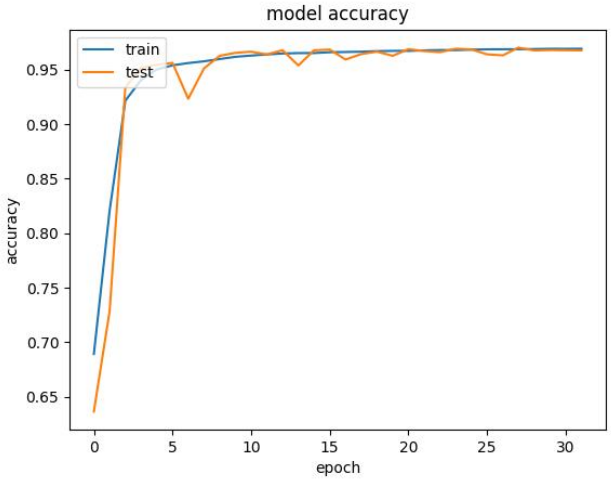

Figure 19: Training vs test accuracy for traffic data on shaped with RF feature importance

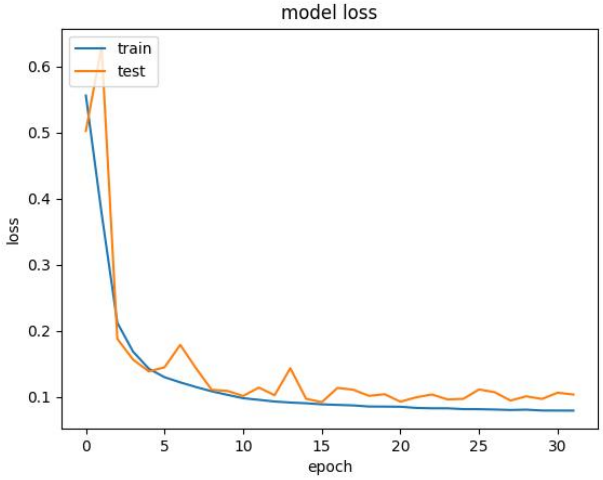

Figure 20: Training vs test loss for traffic data on shaped with RF feature importance.

application categorization as well. ResNet showed the second highest performance in traffic classification when we fed it 9×9 shaped vectors with RF feature weights. However, a simple 9×9 vector outscored all other shapes for application classification in the ResNet experiments.

From the results, there can be seen that at 0% SMOTE level the dataset was in

Table 6: Accuracy for traffic classification

SMOTE level	KNN	GBC	XGBoost	MLP	Resnet
0%	0.886	0.961	0.983	0.832	0.962
20%	0.886	0.961	0.983	0.832	0.957
40%	0.877	0.959	0.982	0.849	0.956
60%	0.869	0.959	0.979	0.789	0.965
80%	0.864	0.957	0.977	0.658	0.952
100%	0.860	0.957	0.974	0.726	0.881

Table 7: F1-score for traffic classification

SMOTE level	KNN	GBC	XGBoost	MLP	Resnet
0%	0.884	0.961	0.983	0.821	0.961
20%	0.884	0.960	0.983	0.821	0.957
40%	0.881	0.960	0.982	0.850	0.956
60%	0.875	0.960	0.980	0.788	0.965
80%	0.864	0.958	0.977	0.676	0.954
100%	0.871	0.958	0.975	0.744	0.892

Table 8: Accuracy for application classification

SMOTE level	KNN	GBC	XGBoost	MLP	Resnet
0%	0.755	0.848	0.896	0.662	0.814
20%	0.750	0.844	0.890	0.629	0.741
40%	0.742	0.840	0.887	0.640	0.804
60%	0.733	0.837	0.886	0.586	0.753
80%	0.731	0.836	0.884	0.571	0.773
100%	0.730	0.834	0.884	0.561	0.800

Table 9: F1-score for application classification

SMOTE level	KNN	GBC	XGBoost	MLP	Resnet
0%	0.750	0.840	0.893	0.591	0.805
20%	0.746	0.840	0.890	0.587	0.739
40%	0.742	0.840	0.888	0.596	0.806
60%	0.736	0.838	0.887	0.558	0.763
80%	0.734	0.837	0.885	0.547	0.781
100%	0.734	0.835	0.885	0.536	0.804

its original form and hence highly imbalanced and as a result of this imbalance the baseline accuracy is higher. Similarly, the baseline accuracy decreases as we move from 0% to 100% SMOTE level. As a result, models with lower SMOTE levels might sometime perform relatively better than the models with high SMOTE levels that can been seen from Table 6, Table 7, Table 8, and Table 9.

4.4 Discussion

Comparing the previous work of [16], [18] and [4] as in Table 10 it is observed that random forest experiment conducted by [4] irrespective of SMOTE level performed better than XGBoost and the same experiment conducted by [16] for Traffic classification. For application classification, XGBoost did show better results compared to CNN [18] but the results produced by random forest [4] showed better performance overall.

Table 10: F1-score compared to previous work

Category	RF [16]	CNN [18]	RF [4]	Our Results for XGBoost
Traffic classification	0.987	–	0.998	0.983
Application classification	–	0.860	0.922	0.893

As seen in Figure 21 the graph shows the traffic and application scores for related work done before compared to our results from XGBoost. Random forest experiments conducted by [4] showed an F1-score of 0.987 and 0.922 for traffic and application classification respectively which was also the highest-performing algorithm for both classes. CNN performed by [18] was not that great at identifying application class compared to the XGBoost performed in this paper. Also, random forest conducted by [16] performed better than XGBoost for traffic classification.

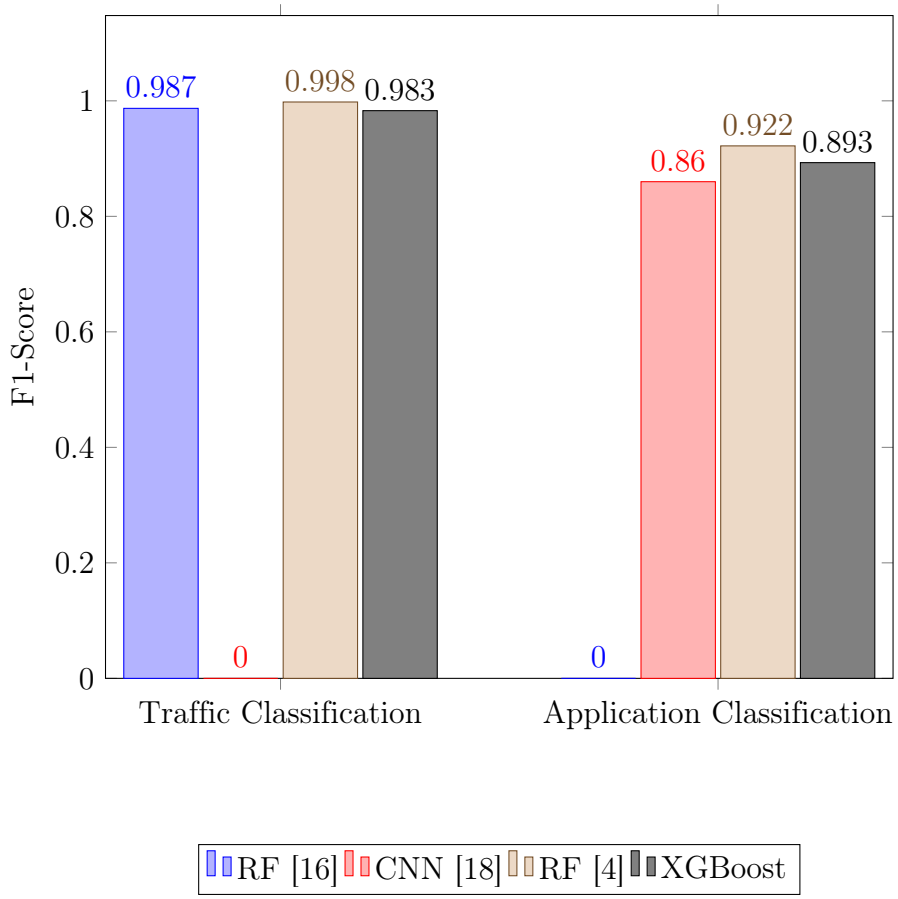

Figure 21: F1-score comparison to related work

CHAPTER 5

Conclusion and Future Work

In our study, we investigated the machine learning algorithms Gradient Boosting, XGBoost, KNN, MLP, and ResNet using the CICDarknet2020 network traffic. For the mentioned multi-class classification task, we categorized traffic and application sample data. For our classification challenge, we performed feature selection and fetched the top 72 features. To determine whether raising SMOTE levels had any impact on the efficiency of the algorithms. Additionally, we computed the ResNet output for three distinct shapes: 9×9, 9×9 random forest weights with spiral, and 9×9 random forest weights derived from [4]. ResNet did not perform better than XGBoost, however, 9×9 with random forest feature weight did the greatest job at classifying traffic.

Finally, our comparison to previous results deduced that though XGBoost is competitive with most of the techniques the results performed by random forest in [4] lead our case for both application and traffic classification. Because there were not enough datasets available, our research was constrained. The most widely utilized and easily accessible dataset for our investigation was CICDarknet2020. In the future, adopting clustering algorithms and then training each cluster to provide samples in place of other oversampling methods may lead to greater performance. It is also possible to employ adversarial obfuscation analysis to understand how various classifiers react to the methods. In order to access the possibilities of a timestamp on the type of traffic and application, time-series constraints could also be utilized. Applications of various other deep learning algorithms could be employed to study the effect of each on the traffic and application category.

LIST OF REFERENCES

[1] NPR, ‘‘Going Dark: The Internet Behind The Internet,’’ 5 2014, (Accessed on 11/25/2022). [Online]. Available: https://www.npr.org/sections/alltechconsidered/2014/05/25/315821415/going-dark-the-internet-behind-the-internet

[2] D. Georgiev, ‘‘How Much of the Internet is the Dark Web in 2022?’’ 11 2022, (Accessed on 11/25/2022). [Online]. Available: https://techjury.net/blog/how-much-of-the-internet-is-the-dark-web/

[3] K. Demertzis, K. Tsiknas, D. Taketzis, C. Skianis, and L. Iliadis, ‘‘Darknet traffic big-data analysis and network management to real-time automating the malicious intent detection process by a weight agnostic neural networks framework,’’ *Electronics*, vol. 10, no. 7, pp. 2--5, 02 2021, (Accessed on 11/25/2022). [Online]. Available: https://www.mdpi.com/2079-9292/10/7/781

[4] N. Rust-Nguyen and M. Stamp, ‘‘Darknet traffic classification and adversarial attacks,’’ *arXiv*, vol. 1, pp. 6--24, 06 2022, (Accessed on 11/25/2022). [Online]. Available: https://arxiv.org/abs/2206.06371

[5] D. Ghimiray, ‘‘The Dark Web Browser: What Is Tor, Is It Safe, and How to Use It,’’ https://www.avast.com/c-tor-dark-web-browser#, 8 2022, (Accessed on 11/25/2022).

[6] Trademark, ‘‘The Tor Project --- History,’’ https://www.torproject.org/about/history/, 8 2022, (Accessed on 11/25/2022).

[7] D. Sarkar, P. Vinod, and S. Yerima, ‘‘Detection of tor traffic using deep learning,’’ in *17th ACS/IEEE International Conference on Computer Systems and Applications*. AICCSA, 11 2020, pp. 1--8, (Accessed on 11/25/2022). [Online]. Available: https://dora.dmu.ac.uk/bitstream/handle/2086/20415/AICCSA_2020_Final_Version_134.pdf?sequence=1&isAllowed=y

[8] R. Dingledine, N. Mathewson, and P. Syverson, ‘‘Tor: The Second-Generation onion router,’’ in *13th USENIX Security Symposium (USENIX Security 04)*. San Diego, CA: USENIX Association, Aug. 2004, pp. 2--5, (Accessed on 11/25/2022). [Online]. Available: https://www.usenix.org/conference/13th-usenix-security-symposium/tor-second-generation-onion-router

[9] C. L. Company, ‘‘Onion router,’’ https://encyclopedia2.thefreedictionary.com/Onion+Router, 2019, (Accessed on 11/25/2022).

[10] Kaspersky, ‘‘What is VPN? How It Works, Types of VPN,’’ https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn, 10 2022, (Accessed on 11/25/2022).

[11] R. Venkateswaran, ‘‘Virtual private networks,’’ *IEEE Potentials*, vol. 20, no. 1, pp. 11--15, 02-03 2001, (Accessed on 11/25/2022). [Online]. Available: https://ieeexplore.ieee.org/document/913204

[12] GeeksforGeeks, ‘‘Virtual Private Network (VPN),’’ https://www.geeksforgeeks.org/virtual-private-network-vpn-introduction/, 9 2022, (Accessed on 11/25/2022).

[13] L. A. Iliadis and T. Kaifas, ‘‘Darknet traffic classification using machine learning techniques,’’ in *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. MOCAST, 2021, pp. 1--4, (Accessed on 11/25/2022).

[14] A. Habibi Lashkari., G. Draper Gil., M. S. I. Mamun., and A. A. Ghorbani., ‘‘Characterization of tor traffic using time based features,’’ in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP,*, INSTICC. SciTePress, 2017, pp. 253--262, (Accessed on 11/25/2022). [Online]. Available: https://pdfs.semanticscholar.org/d76f/32eb3af1a163c0fde624e9fc229671ca75b6.pdf

[15] Y. Hu, F. Zou, L. Li, and P. Yi, ‘‘Traffic classification of user behaviors in tor, i2p, zeronet, freenet,’’ in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 12 2020, pp. 418--424, (Accessed on 11/25/2022). [Online]. Available: https://ieeexplore.ieee.org/document/9343185

[16] H. Karagöl, O. Erdem, B. Akbas, and T. Soylu, ‘‘Darknet traffic classification with machine learning algorithms and smote method,’’ in *2022 7th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 09 2022, pp. 374--378, (Accessed on 11/25/2022). [Online]. Available: https://ieeexplore.ieee.org/document/9919462

[17] S. Sridhar and S. Sanagavarapu, ‘‘Darknet traffic classification pipeline with feature selection and conditional gan-based class balancing,’’ in *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)*. IEEE, 11 2021, pp. 1--4, (Accessed on 11/25/2022). [Online]. Available: https://ieeexplore.ieee.org/document/9685743

[18] A. Habibi Lashkari, G. Kaur, and A. Rahali, ‘‘Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning,’’ in *2020 the 10th International Conference on Communication and*

Network Security, ser. ICCNS 2020, Tokyo, 2021, p. 1–13, (Accessed on 11/25/2022). [Online]. Available: https://doi.org/10.1145/3442520.3442521

[19] KDnuggets, ‘‘Most Popular Distance Metrics Used in KNN and When to Use Them,’’ 2022, (Accessed on 11/25/2022). [Online]. Available: https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html

[20] GeeksforGeeks, ‘‘Residual Networks (ResNet) - Deep Learning,’’ https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/, 8 2022, (Accessed on 11/25/2022).

APPENDIX

Features Used for Analysis

Table A.11: Features selected after data preprocessing

Flow Duration	Bwd Packets/s
Total Fwd Packet	Packet Length Min
Total Bwd packets	Packet Length Max
Total Length of Fwd Packet	Packet Length Mean
Total Length of Bwd Packet	Packet Length Std
Fwd Packet Length Max	Packet Length Variance
Fwd Packet Length Min	FIN Flag Count
Fwd Packet Length Mean	SYN Flag Count
Fwd Packet Length Std	RST Flag Count
Bwd Packet Length Max	PSH Flag Count
Bwd Packet Length Min	ACK Flag Count
Bwd Packet Length Mean	URG Flag Count
Bwd Packet Length Std	CWE Flag Count
Flow Bytes/s	ECE Flag Count
Flow Packets/s	Down/Up Ratio
Flow IAT Mean	Average Packet Size
Flow IAT Std	Fwd Segment Size Avg
Flow IAT Max	Bwd Segment Size Avg
Flow IAT Min	Fwd Bytes/Bulk Avg
Fwd IAT Total	Fwd Packet/Bulk Avg
Fwd IAT Mean	Fwd Bulk Rate Avg
Fwd IAT Std	Bwd Bytes/Bulk Avg
Fwd IAT Max	Bwd Packet/Bulk Avg
Fwd IAT Min	Bwd Bulk Rate Avg
Bwd IAT Total	Subflow Fwd Packets
Bwd IAT Mean	Subflow Fwd Bytes
Bwd IAT Std	Subflow Bwd Packets
Bwd IAT Max	Subflow Bwd Bytes
Bwd IAT Min	FWD Init Win Bytes
Fwd PSH Flags	Bwd Init Win Bytes
Bwd PSH Flags	Fwd Act Data Pkts
Fwd URG Flags	Fwd Seg Size Min
Bwd URG Flags	Src Port
Fwd Header Length	Dst Port
Bwd Header Length	Label
Fwd Packets/s	Label.1