Network Traffic Analysis Using Machine Learning

Oindri Kar
School of Engineering
University of Guelph
Ontario, Canada
okar@uoguelph.ca

Niki Esmaeili
School of Engineering
University of Guelph
Ontario, Canada
esmaeiln@uoguelph.ca

***Abstract*— In the age of increasingly digital communications, the growth of automated work and rapidly connected devices is driving demand for a powerful network. Existing networks are finding it increasingly difficult to handle the increase in traffic generated by these technologies. Enter the topic of Software Defined Networking (SDN), a simple and flexible solution that incorporates machine learning (ML) to operate in networking. SDN has become a symbol of innovation that solves the inefficiencies of traditional networking, including network management and distribution issues. The data collected by SDN controllers is an asset for analytics, paving the way for the application of ML models that will revolutionize network management. This project focuses on the use of network data and proposes simple ML-based techniques for classifying network traffic and integrating these models into a software-defined ecosystem.**

Keywords— Network Traffic, Machine Learning, Deep Learning, Software Defined Networking, Classification

I. INTRODUCTION

In recent academic research, machine learning (ML) techniques are effective in analyzing and classifying network connections, which is very important in today's network infrastructure environment. Despite this progress, the transition from theoretical models to practical applications presents different challenges. Evaluating the effectiveness of these methods in real-world situations often presents many issues that are not apparent during regulatory evaluation. Identifying and resolving these issues is important for the development of ML-based network analysis in manufacturing facilities. The main factors for evaluating the effectiveness of machine learning (ML) for traffic classification include the ability of the ML algorithm in classification, correctness, and correctness of the effect distribution of results, as well as the reliability and representativeness of the data used during the study. These characteristics play an important role in determining the usefulness of the ML algorithm as a reliable traffic classification tool [1]. In general, network traffic distribution methods are divided into three types: port-based, payload-based, and machine learning-based. Port-based methods rely on the assumption that TCP or UDP port numbers are always used, but the reliability of this method is affected by the use of non-standard and strong ports. Payments-based classifications, particularly data analytics (DPI), analyze payment packages to identify specific patterns or key elements. However, DPI suffers from encrypted payments and requires high computing power. As a result, machine learning-based methods have been developed as a solution, providing a practical method that can be effective for both encrypted and unencrypted traffic. These approaches often employ traditional machine learning (ML) algorithms, like K-Nearest Neighbors (KNN). Nonetheless, the efficacy of these classical ML techniques hinges on features crafted by humans, constraining their ability to generalize across diverse scenarios [2].

TABLE I. SUMMARY OF TRAFFIC CLASSIFICATION APPROACHES[2]

Methods	Description	Advantages	Disadvantages
Port-Based	Classifies Packets by Port numbers	Fast, Low resource-consuming, High accuracy	Does not implement the application Layer payload, infeasible for Hidden ports
Payload-Based	Uses Deep Packet Inspection to look into packet contents	Handle Services with dynamic ports	High computational cost, Inapplicable for encrypted data and Privacy issues
ML-Based	Extract features from packet payload or statistical characteristic with ML model	Handle dynamic port and encrypted data, Fast Technique compared to Deep packet inspection classification	Longer classification time compared with port-based method

Our methodology includes evaluating ML-based techniques and comparing relevant data, with a focus on the use of the K-Nearest Neighbor (KNN) algorithm for traffic classification. We address the challenges associated with developing and deploying data for training ML models such that their performance in real-time traffic is affected by measurements found offline. Our study highlights the need for a predefined methodology for accurate and precise identification. The study of machine learning algorithms forms an important part of our research, with the choice of algorithm determined by the specific properties and needs of traffic distribution. To evaluate the performance of these algorithms, we use the confusion matrix and evaluation methods that include accuracy, precision, recall, and F1 score.

Our goal is to use visualization to clearly and instinctively identify subjects seen through the screen. Fundamentally, our research represents a decision to go beyond simple traffic analysis and provide communication organizations with good insights resulting from understanding the quality of traffic distribution.

The following is how the paper's structure unfolds:
Section I: Following the Introduction, this part outlines the Project Motivation and Objectives.
Section II: Provides a concise summary of previous work in the field of Network Traffic Classification
Section III: Reveals the Project Methodology, explaining the technique utilized in implementing the code.
Section IV: Comprises the outcomes of the ML-algorithm's implementation.
Section V: Conducts a comparative analysis based on the algorithms used.
Section VI: delves into Future Directions, imagining potential avenues for future research.
Section VII: To conclude the discussion, the author summarizes major insights and conclusions garnered from the research.

II. LITERATURE REVIEW

Review of Existing Research:

Recently, many researchers have introduced different methods to distribute network connections. In this section, we'll look at some network traffic distribution techniques. Three simple methods can be used for this purpose. That is, classification based on port classification, classification based on payment content classification, and classification based on machine learning classification [3].

In port-based techniques, the identification of network traffic relies on the use of these numerical identifiers. These identifiers are distributed by the Internet Assigned Numbers Authority (IANA). Within this framework, various applications on a network's local host utilize IANA-assigned port numbers as a nexus for communication, allowing other hosts to connect using this established point. To identify the server's end of a new TCP connection between a client and server, a network classifier is required to inspect TCP SYN packets. This inspection is part of the initial phase of the TCP three-way handshake, which is essential for setting up a session. The destination port number in the TCP SYN packet facilitates the routing of the application to the corresponding port [4]. Regrettably, these methods are impaired by certain limitations: With the expansion of applications, the utilization of port numbers can become erratic [5]. These techniques may not be suitable when some applications don't register their port numbers with IANA [4]. If your application uses dynamically allocated ports, the results of this process will be affected.

A payment-based approach, commonly called deep packet inspection (DPI), analyzes data packets by comparing known patterns or signatures associated with applications in network traffic. Most of these payment methods include checking the contents of the package and verifying its signature on file. This method provides more accurate results than the port method and is particularly useful for peer-to-peer (P2P) transactions. Despite their accuracy, these methods have drawbacks and disadvantages. The implementation of these methods is sophisticated and incurs substantial computational expenses, leading to an increased processing demand on the identification hardware. Furthermore, utilizing this approach with encrypted traffic can be challenging or even infeasible. Additionally, the scrutiny of packet contents raises concerns about infringements of privacy norms and regulations [6].

ML Algorithm-Based Research: Machine learning-based approaches can avoid some of the pitfalls associated with port-based and payload-based systems. Using machine learning for traffic classification will reduce the computational cost and make it easier to identify incoming traffic [11].

Gaps in Literature:

Despite development, there are still gaps about:

- Limited Comparative Analysis: There is a requirement for detailed comparisons of method performances.
- Dataset Diversity: Inconsistency in dataset types necessitates study across several features.
- Interpretability: Improve the interpretability KNN.
- Dynamic Feature relevance: Examining how feature relevance changes over time.

III. METHODOLOGY

A. DATA COLLECTION

The dataset has been taken from Kaggle. Kaggle is a popular platform for data science and machine learning practitioners that offers datasets, competitions, kernels (code notebooks), and a community of data enthusiasts. Computer Network Traffic Dataset - This is a CSV file of approximately 500K in size, summarizing historical network traffic data. Containing around 21,000 entries, it documents the activity of 10 local workstations over three months. During this time, half of

these local IP addresses were compromised and subsequently integrated into different botnets. [7]

Each row consists of four columns:

- date: yyyy-mm-dd (from 2006-07-01 through 2006-09-30)
- l_ipn: local IP (coded as an integer from 0-9)
- r_asn: remote ASN (an integer which identifies the remote ISP)
- f: flows (count of connections for that day)

Reports of "odd" activity or suspicions regarding machine's behaviour triggered investigations on the following days (despite the possibility that the machine may have been compromised before this)

Date : IP
08-24 : 1
09-04 : 5
09-18 : 4
09-26 : 3 6

[7] This public dataset was found on
http://statweb.stanford.edu/~sabatti/data.html

B. EXPLORATORY DATA ANALYSIS

Data visualization in network traffic analysis enhances insights by graphically representing patterns and trends. Visualizations like Scatter plots, bar graphs and Heatmap help in identifying correlations and key indicators. Overall, visualizations of data facilitate a more intuitive understanding of data, aiding decision-making in mitigating network traffic.

1. Scatter Plot

The scatter plot is created using 'Packets' and 'Flow Duration' from the network traffic dataset displaying the relationship between these two variables for each data point in the dataset.

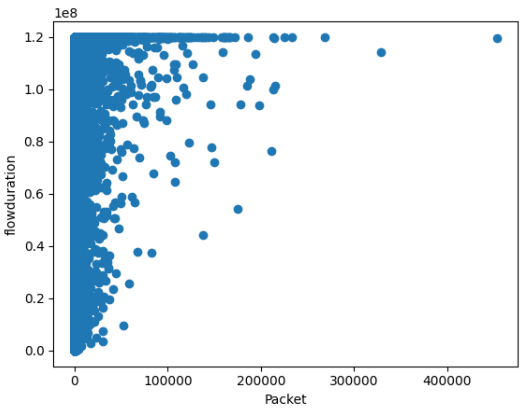

Fig 1 : Scatter plot of dataframe

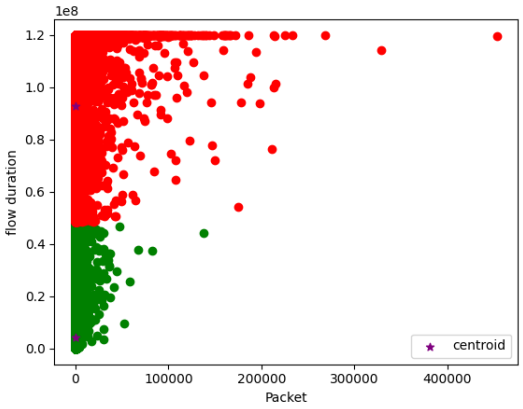

Fig 2 : Scatter plot of dataframe after clustering

Fig 1 displays the scatter plot of the dataframe prior to clustering. Subsequently, clustering is applied using the KMeans algorithm. KMeans clustering is an unsupervised machine learning technique that divides a dataset into K unique, non-overlapping subsets or clusters.

After KMeans we got the output 0 and 1:
0 is for dataframe1 and marked in green colour
1 is for dataframe2 and marked in red colour

2. Bar graph

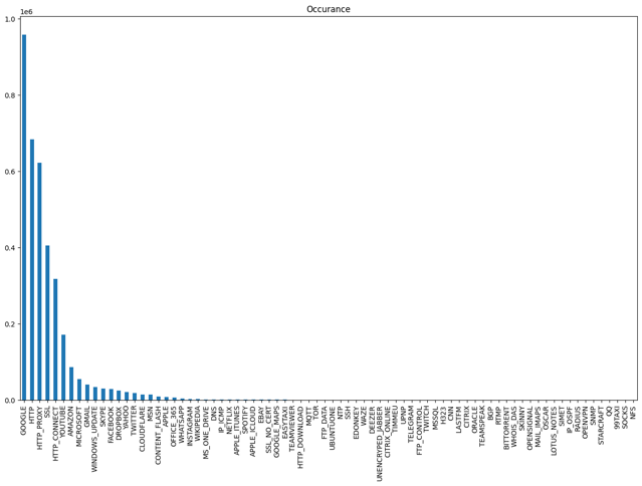

Fig 3: No of records of each application

Fig shows the plot for the number of records for individual applications using a bar plot based and their occurrence counts in the DataFrame. It calculates the occurrence counts of each unique protocol name and creates a bar plot displaying these counts.

3. Line Plot

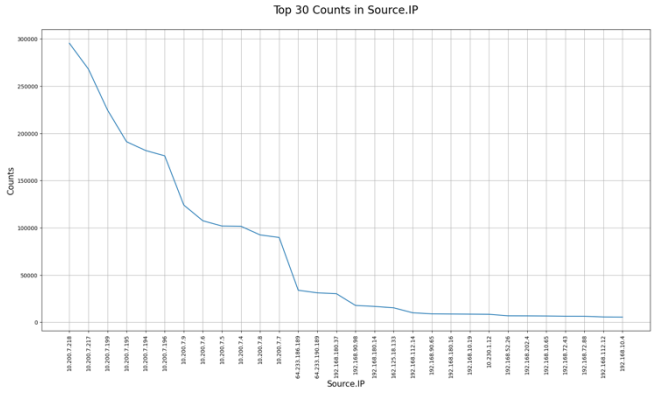

Fig5: Line Plot of Destination IP

This Fig shows the line plot for the top 30 counts in the Destination IP' column. This shows the distribution of counts for the top 30 ‘Destination IP' values in the dataset.

4. Heatmap

Heatmaps represent data graphically, with raster values indicated by color. Particularly for confusion matrices, heatmaps provide a visual way to understand and explore patterns in the data.

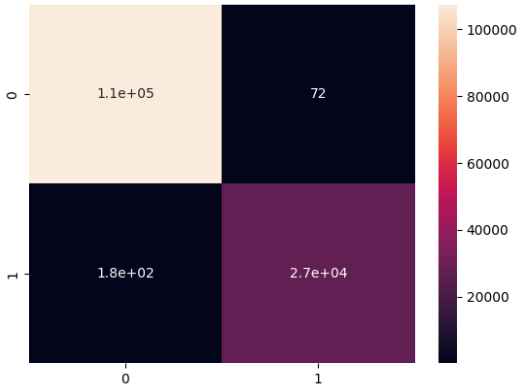

Fig 6 : Heatmap visualization of confusion matrix

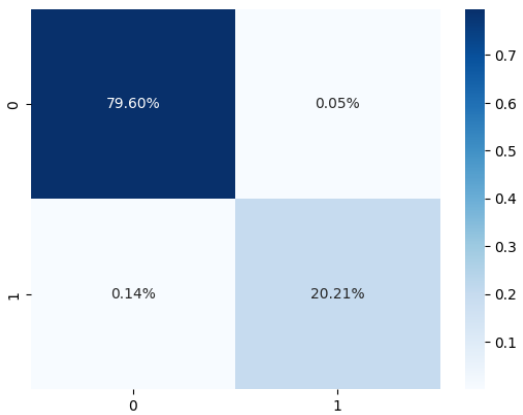

Fig 7 : Heatmap visualization of normalized confusion matrix

- The confusion matrix (cm) holds details on true positives, true negatives, false positives, and false negatives across various categories of network traffic.
- The heatmap here shows a visual representation of this information by using a colour gradient to represent the values in the matrix. Lighter or darker shades of colour indicate higher or lower values, respectively.
- Annotations within the cells provide precise numerical information about the classification results, aiding in understanding the distribution of correct and incorrect predictions.
- The heatmap depiction of the normalized confusion matrix helps in grasping the model's effectiveness by showing the normalized figures for true positive, false positive, true negative, and false negative classifications across diverse classes or types of network traffic.

This visualization helps quickly assess the model's performance, identify areas of correct or incorrect classification, and gain insights into the strengths and weaknesses of the classification model for network traffic.

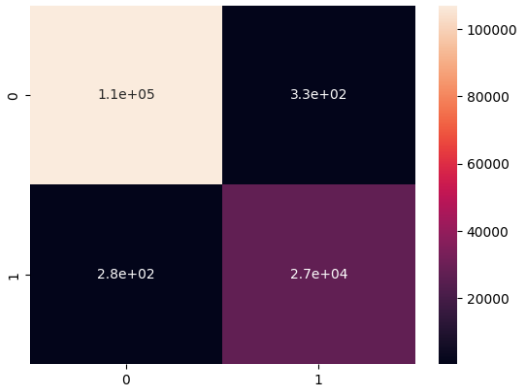

Fig 8: Heatmap visualization of confusion matrix after GRU

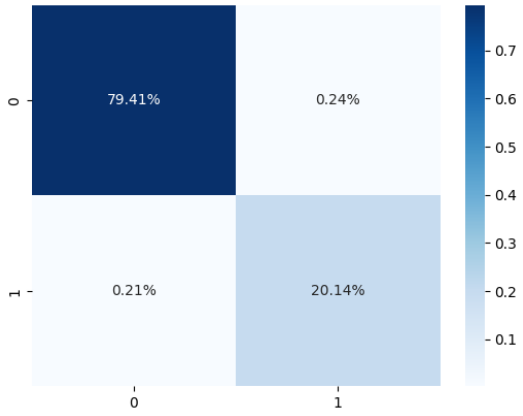

Fig 9 : Heatmap visualization of normalized confusion matrix after GRU

These heatmap visualizations display (fig & fig) the classification results from the GRU model, indicating the false positive, true positive, true negative, and false negative predictions across various categories of network traffic.

Gated Recurrent Units (GRU) is a category of Recurrent Neural Network (RNN) architecture that is widely used in sequence modelling, including network traffic analysis.

C. DATA PREPROCESSING

Data preprocessing assumes a pivotal role in refining raw datasets. The ensuing steps delineate a systematic approach to enhance data quality and facilitate downstream analytical processes [8].

i. Handling Missing Data: Identification and resolution of missing data are imperative. Robust strategies, such as removing rows with missing values or imputing them with statistically sound measures like the mean, are employed to ensure data completeness. We checked if any column in the dataframe had null values which were replaced with the mean of values.
ii. Handling Categorical Variables: To incorporate categorical variables into analytical models, a judicious application of label encoding is executed This method methodically converts categorical data into a numerical format, thereby rendering them suitable for use with machine learning algorithms.
iii. Standardization: The standardization of numerical features is pivotal for achieving uniformity in data ranges. Leveraging standard scalar techniques, numerical attributes are scaled to a common standard, mitigating the impact of disparate magnitudes, and fostering equitable contribution to subsequent analyses.
iv. Data Splitting: split the dataset into training and testing sets, allowing for the training of models on a portion of the data and assessment of their effectiveness on data that has not been previously seen.

In summary, these meticulous data preprocessing steps contribute to the refinement of datasets, laying the groundwork for precise and reliable analyses.

D. FEATURE SELECTION

The column which had no significant contribution to the prediction of the network analysis was dropped.

Feature Engineering:

Specific columns named 'Source.Port', 'Destination.Port', 'L7Protocol', and 'Protocol' are dropped from the dataset. Then the remaining columns are displayed after removal has been performed. This structured approach is useful for removing columns that might not contribute to the analysis, such as categorical columns or columns containing irrelevant or redundant information, thus preparing the dataset for further analysis or model building. This ensures the uniform preprocessing of diverse data types, fostering the creation of resilient machine-learning models.

E. EXPERIMENTAL SETUP

To enhance model robustness and mitigate overfitting, a cross-validation strategy was employed during the model training phase. Specifically, k-fold cross-validation with k=5 was utilized. This involves dividing the training into five parts, where the model is trained on four of the five parts of the data and is validated on the fifth part. This procedure was performed five times and model performance was averaged between iterations [9].

Evaluation Metrics Chosen:

The models were evaluated using a comprehensive set of metrics to provide a nuanced understanding of performance.

- Accuracy: Check the accuracy of the forecast model and its suitability for the data balance.
- Precision: It assesses the accuracy of positive predictions, emphasizing the reliability of the model when it predicts positive instances.
- Recall (Sensitivity): It quantifies the ability of the model to identify correctly of all relevant instances, particularly relevant in scenarios where false negatives are costly.
-

F. IMPLEMENTED ML ALGORITHM

K Nearest Neighbors (KNN):

The K-Nearest Neighbors algorithm stands as a versatile and intuitive approach in the realm of supervised machine learning, seamlessly addressing both regression and classification tasks. This approach determines the identity or value of new data by examining the features or values of its nearest neighbors in the training data. Here, “k” is an important parameter that represents the number of neighbors considered in the decision process [10].

K-Nearest Neighbors (KNN) proves to be an apt choice for network traffic prediction for several reasons. Its fundamental straightforwardness and ease of use render it approachable, allowing for uncomplicated implementation and understanding, even for those with limited machine learning knowledge. Additionally, KNN is particularly adept at managing the non-linear relationships frequently observed in customer churn cases, as it doesn't presuppose any specific functional form.

Testing the accuracy of the classifier is important to assess the correctness of the choices made by the machine learning algorithm when applied to new, unseen data. To this end,

various metrics have been developed by exploiting the results of the classification process of machine learning classifiers.

The most frequently used performance metrics for binary classification problems are:[11]

- Accuracy: (TP+TN) / (TP+TN+FP+FN). The ratio of correct predictions to the total number of predictions made.
- Precision: TP / (TP+FP). The ratio of the total number of correctly classified items to the total number of predicted positive items.
- Recall: TP / (TP+FN). The ratio of total number of correctly classified items to the total number of positive items.

As mentioned earlier we have used Gated Recurrent Unit (GRU) to calculate the accuracy, sensitivity, and Precision. When using GRU in network traffic dataset we used the preprocessed data, then formatted it into sequences and fed it to the GRU architecture. The GRU model was then trained to learn from this sequential data to perform tasks to calculate accuracy, sensitivity, and precision. GRU is effective in capturing temporary dependencies in sequential data like network traffic. It potentially yields better accuracy by understanding the temporal patterns and long-range dependencies in the traffic data.

IV RESULTS

	Accuracy	**Sensitivity/ Recall**	**Precision**
KNN method	0.97906195	0.993609274	0.98005378
Using GRU	0.99658822	0.998184155	0.99753392

The results show that the Gated Recurrent Unit gives better accuracy, sensitivity/recall and precision as compared to K Nearest Neighbor method.

V FUTURE SCOPE

Security Enhancements: Improved machine learning algorithms will focus on detecting minor abnormalities and behavioral changes in network traffic patterns, helping in the identification of zero-day attacks and sophisticated threats. AI-powered systems will evolve to autonomously respond to cyber threats by implementing adaptive security measures, containment, and mitigation strategies in real-time scenarios.

AI/ML-driven Network Optimization: Artificial Intelligent algorithms will optimize traffic routing in complex networks, dynamically allocating resources and paths based on real-time demands and traffic patterns. Advanced analytics will predict network failures and performance degradation, enabling proactive maintenance and minimizing downtime.

5G and IoT Integration: AI will assist in creating efficient and customized network slices in 5G networks, ensuring tailored services for diverse applications and users. ML-based anomaly detection and behavior analysis will safeguard IoT networks by identifying abnormal device behaviors and potential security breaches.

Privacy and Compliance: Development of privacy-centric AI models and encryption methods to ensure compliance with privacy regulations while still enabling effective network traffic analysis.

Explainable AI(XAI) in Network Analysis: Integration of explainable AI techniques to make network traffic analysis models more interpretable, enabling users to understand the rationale behind model decisions and predictions.

Edge Computing and Distributed Networks: Implementation of AI-based traffic analysis at the network edge to handle decentralized architectures like edge computing and IoT, reducing latency and bandwidth usage.

Cross-domain Integration: Collaboration between network traffic analysis and other domains like healthcare, transportation, and smart cities to enhance communication, security, and data analytics across interconnected systems.

Ethical Considerations: Development and implementation of ethical guidelines and governance frameworks for deploying AI in network traffic analysis, ensuring fairness, transparency, and accountability in decision-making processes.

Overall, the future of network traffic analysis will revolve around implementing AI, ML, and advanced analytics to enhance security, optimize performance, and address the challenges posed by evolving network infrastructures and emerging technologies. This evolution will encompass a wide range of technological advancements, ethical considerations, and interdisciplinary collaborations to drive innovation and reliability in network traffic analysis systems.

VI CONCLUSION

In this project we have explored Machine Learning (ML) algorithm within the domain of network traffic classification. Our project on network traffic analysis using the K-Nearest Neighbors (KNN) algorithm has provided valuable insights into understanding and analyzing network traffic patterns. Through our analysis, we observed that KNN, while a simple and intuitive algorithm, demonstrated effectiveness in certain scenarios for classifying network traffic based on its nearest neighbors in the feature space. Moving forward, further exploration involving different algorithms and more advanced techniques could provide a more comprehensive understanding and improved accuracy in analyzing network traffic.

REFERENCES

[1] M. Ramires, A. S. Gomes, S. Rito Lima and P. Carvalho, "Network Traffic Classification using ML: A Comparative Analysis," *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, Madrid, Spain, 2022, pp. 1-6, doi: 10.23919/CISTI54924.2022.9820583.

[2] S. M. Rachmawati, D. -S. Kim and J. -M. Lee, "Machine Learning Algorithm in Network Traffic Classification," *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Republic of, 2021, pp. 1010-1013, doi: 10.1109/ICTC52510.2021.9620746.

[3] Noora AI Khater and Richard E Overill," Network Traffic Classification Techniques and Challenges", The Tenth International Conference on Digital Information Management (ICDIM), 2015, pp 43-48.

[4] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning", IEEE Communications Surveys & Tutorials, Vol. 10, No. 4, fourth quarter 2008, pp 56-76.

[5] T.Karagiannis, A. Broido, N.Brownlee, and K.Claffy, " Is P2P dying or just hiding?", Proc. 47th annual IEEE Glibal Telecommunications Conferene(Globecom 2004), Dallas, Texas, USA, November,December 2004.

[6] Zhong Fan and Ran Liu," Investigation of Machine Learning Based Network Traffic Classification", International Symposium on Wireless Communication Systems (ISWCS) 2017, pp1-6.

[7] Reference: This public dataset was found on http://statweb.stanford.edu/~sabatti/data.html

[8] V. Chang, X. Gao, K. Hall and E. Uchenna, "Machine Learning Techniques for Predicting Customer Churn in A Credit Card Company," 2022 International Conference on Industrial IoT, Big Data and Supply Chain (IIoTBDSC), Beijing, China, 2022, pp. 199-207, doi: 10.1109/IIoTBDSC57192.2022.00045.

[9] P. Gopal and N. B. MohdNawi, "A Survey on Customer Churn Prediction using Machine Learning and data mining Techniques in E-commerce," 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Brisbane, Australia, 2021, pp. 1-8, doi: 10.1109/CSDE53843.2021.9718460.

[10]M. H. Seid and M. M. Woldeyohannis, "Customer Churn Prediction Using Machine Learning: Commercial Bank of Ethiopia," 2022 International Conference on Information and Communication Technology for Development for Africa (ICT4DA), Bahir Dar, Ethiopia, 2022, pp. 1-6, doi: 10.1109/ICT4DA56482.2022.9971224

[11] Y. D. Goli and R. Ambika, "Network Traffic Classification Techniques-A Review," *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Belgaum, India, 2018, pp. 219-222, doi: 10.1109/CTEMS.2018.8769309.

U. Baek, B. Kim, J. Park, J. Choi and M. Kim, "MISCNN: A Novel Learning Scheme for CNN-Based Network Traffic Classification," *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Takamatsu, Japan, 2022, pp. 01-06, doi: 10.23919/APNOMS56106.2022.9919961.

[12]M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 2016, pp. 2451-2455, doi: 10.1109/CompComm.2016.7925139.

[13]N. Al Khater and R. E. Overill, "Network traffic classification techniques and challenges," 2015 Tenth International Conference on Digital Information Management (ICDIM), Jeju, Korea (South), 2015, pp. 43-48, doi: 10.1109/ICDIM.2015.7381869.

[14]Y. D. Goli and R. Ambika, "Network Traffic Classification Techniques-A Review," 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 2018, pp. 219-222, doi: 10.1109/CTEMS.2018.8769309.

[15]G. Szabo, I. Szabo and D. Orincsay, "Accurate Traffic Classification," 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Espoo, Finland, 2007, pp. 1-8, doi: 10.1109/WOWMOM.2007.4351725.

[16]Y. Xue, D. Wang and L. Zhang, "Traffic classification: Issues and challenges," 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, USA, 2013, pp. 545-549, doi: 10.1109/ICCNC.2013.6504144.

[17]S. Patel, A. Gupta, Nikhil, S. Kumari, M. Singh and V. Sharma, "Network Traffic Classification Analysis Using Machine Learning Algorithms," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 1182-1187, doi: 10.1109/ICACCCN.2018.8748290.

[18]https://bmsit.ac.in/public/assets/pdf/ece/research/Research_Compendium_2018-19%20Dept%20ECE%20%2027.08.2020.pdf?cv=1