TECHNIUM

Technium Vol. 13, pp.12-26 (2023)
ISSN: 2668-778X
www.techniumscience.com

Performance Evaluation of Machine Learning Nave Bayes Algorithms for Network Traffic Classification

Afrah Salman Dawood

University of Technology- Iraq

Abstract: Network Traffic Classification (NTC) is an important field for different network statistics like management, malware detection and other paramount constraints. Artificial Intelligence (AI) including Machine Learning (ML) and Deep Learning (DL), on the other hand, plays a very important field nowadays due to its significant capabilities with an extremely different fields and for complex problems. ML, specifically, provides tools in the most important network fields like traffic management, security, etc. Performance evaluation is a very important aspect of any system. This research paper provides a method for NTC using ML Nave Bayes (NB) algorithm in terms of Bernoulli, Multinominal and Gaussian for classifying captured network traffic in two different datasets and perform a performance evaluation and a comparison among these algorithms. The first dataset is a VPN-nonVPN (ISCXVPN2016) dataset while the second is a packet-captured regular Wi-Fi traffic flow dataset for video browsing on the web. Results were comparable in terms of f1-score, accuracy and processing time. Bernoulli NB provides average 93.05% accuracy with 742 ms, Multinominal NB provides average 98.78% accuracy with 78.3 ms processing time and finally, Gaussian NB provides average 69.14% with 46.85 ms processing time.

Keywords: Network traffic classification, Machine learning, Nave Bayes, Bernoulli, Multinominal, Gaussian, Performance evaluation.

1. INTRODUCTION

Network traffic classification (NTC) in network engineering represents an important process for categorizing network traffic in accordance with different parameters (i.e., port number, protocol, etc.). It presents a way for quantifying, monitoring and understanding network traffic in order to troubleshoot network issues for Internet Service Provider (ISP), in other words, it is used to estimate network system's capacity based on Quality of Service (QoS) of traffic flow or lawful interception. The development of communication systems and networks concerning the quantity of clients and the amount of created traffic, presents various daily difficulties to Network Traffic Monitoring and Analysis (NTMA) [1], counting: (1) putting away and dissecting traffic information, (2) involving traffic information for business objectives through

acquiring knowledge, (3) traffic information combination, (4) traffic information approval, (5) traffic information security, and (6) traffic information obtaining. . performance evaluation accompanies various features, the first is the improvement of scientific apparatuses and simulation models to decide the exhibition or accuracy of the network or algorithm and the other aspect of performance evaluation is the utilization of simulation models or estimation campaigns in either real network conditions or research facility testbeds to assess specific performance proportions of a current or future radio organization innovation. The phenomenal expansion in the quantity of associated hubs and the volume of information enhance the organization intricacy, calling for proceeding with studies to examine and screen the networking performance [2], [3]. Moreover, the accessibility of huge and heterogeneous measure of traffic information requires taking on new methodologies for observing and investigating the network management data. Because of these difficulties, most works center explicitly around one part of NTMA, e.g., anomaly detection, traffic arrangement, or QoS. One can significantly profit from the data that is acquired from the network traffic matrix in tasks like routing setup, asset designation, load adjusting, failure identification, traffic scheduling and re-routing, power utilization minimization, and others. NTC turns into an extremely difficult issue for clients and service providers Traditional NTC methods for NTMA depend on port, stream static and payload investigation. The port-based strategy which utilizes TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) port numbers is one of the quickest and most straightforward way to deal with nonstop network monitoring. The payload-based conspire examines the parcel's payload to recognize the mark or grammar of every application's bundle payload. Nonetheless, these plans have their weaknesses as far as managing a lot of information and various kinds of network traffic. With the fast advancement of AI, many explorations work research and apply the flow static-based technique [4] [5] to classify the network traffic. This approach depends on the data acquired from the packets, for example, the quantity of bytes in packet payload, packet time period, course of the packet, and so on. After that, these highlights are prepared in the AI calculations to arrange the network traffic into various sorts of administrations. The flow static-based technique can manage the issues of port and payload-based strategies including the intangibility of payload or dynamic port. A new and important network transport protocol QUIC (Quick UDP Internet Connections) developed by Google and run on the top of UDP. Its important feature is eliminating the requirement of the initial TCP handshake process and add its own encryption system. The performance evaluation of QUIC has been concentrated on by a couple of examination papers in the writing, primarily concentrate in on the utilization instance of QUIC as a transport for HTTP [6].

Machine Learning (ML) and Deep Learning (DL), on the other hand is a subset of Artificial Intelligence (AI) which refers to the recreation of human knowledge in machines that are customized to think like people and copy their activities and includes any technique that is able to mimic human behavior. The accuracy of AI models is noticeable but it is not the only aspect that is of utmost importance. For sensitive domains, a detailed understanding of the model and the outputs is also important. The underlying ML and DL algorithms construct complex models that are opaque for humans. Specifically, ML permits software applications to turn out to be more precise at anticipating results without being explicitly programmed to do as such. ML algorithms utilize historical information as con-tribution to anticipate new result values. Naive

Bayes (NB) [7] is an ML model that is used for large volumes of data and it is a theorem that works on conditional probability. The NTC is better anticipated from its statistical qualities caught from constant traffic grid, which is alluded to as network traffic matrix expectation. The traffic grid is treated as a period series and the expectation is applied to the verifiable traffic streams. To accomplish this, scientists have applied ML strategies on strong related information in network traffic that has long-range reliance highlight [8].

In this paper, **a new packet-based method has been proposed** based on Nave Bayes (NB) classification (Bernoulli, Multinominal and Gaussian), which is set of supervised learning methods for multi-class classification, to classify two types of network traffic flow for different media in two different datasets. In this scenario, a Bernoulli NB, Multinominal NB and a Gaussian NB models, which implement the training and classification for data that is distributed according to multivariate distributions, were implemented. Then these models were trained and tested for a VPN-nonVPN network traffic flow dataset and a video browsing stream on Wi-Fi traffic flow dataset. The code is implemented in TensorFlow with python programming language run on jupyter notebook platform.
The rest of this research is organized like follows: section **2** explains several related works and theoretical background of NB models that is concerned with ML. Section **3** is about characteristics of different service protocols that is going to be classified in this research. Section **4** explains the proposed method based on Bernoulli, multinominal and Gaussian NBs including details on training and testing datasets with TensorFlow and python programming for the proposed models. Section **5** presents experimental results and discusses several regarding subjects. Finally, section **6** highlights the conclusion and future ideas of the worked model.

2. BACKGROUND AND RELATED WORK

2.1. Related Work

This section presents some of the current researches related to the topic of network traffic classification in the scope of ML and DL. In addition, the motivation for a network traffic flow classification using TensorFlow features with varied dataset and NB multinominal.
To solve the problem with Google's QUIC (Quick UDP Internet Connection) Protocol, Tong, et al. [9] put out a brand-new method for categorizing media streams based on convolutional neural networks (CNNs). Because operators' visibility into network traffic is reduced by this Protocol, it has a lot of problems classifying network traffic. The model consists of two multi-stage classification stages to describe Google Hangout's voice and text chat, followed by a second multi-stage classification step for network flows into file transfer, streaming video, or Google Play music. Their method achieves F1-score of 99.24%.
Li, et al. [10] introduced a novel Recurrent Neural Network (RNN) multi-classifying system for network traffic classification by designing the Byte Segment Neural Network (BSNN) model. The model is made up of a softmax function to determine the expected class of data, a segment generator, an attention encoder, and a representation vector. They used their model to classify data from five distinct protocols, including QQ, PPLive, DNS, 360, and BitTorrent, on real-world data. and gained an F1-measure of about 95.82 percent.

Chen, et al. [11] proposed a network traffic classification model NTCNET based on CNNs. They used dataset to implement simulation experiments, and compare the test results with a variety of traditional classification methods. Their results showed that their module NTCNET has better precision, robustness and accuracy, with an accuracy of 99.66%.
Lim, et al. [12] introduced a five DL models using CNN and ResNet for network classification. The performance of network traffic classification has been analyzed for packet-based datasets according to f1 score and demonstrated their effectiveness.
In order to propose three new attacks to create ANT, Sadeghzadeh, et al. [13] evaluated the robustness of DL-based network traffic classifiers with Adversarial Network Traffic (ANT) and proposed three input space categories in DL-based network traffic classification, including packet, flow content, and flow time series classifications. They use One-Dimensional CNN (1DCNN) to classify their dataset (ISCXVPN2016) which includes network traffic from Skype, Facebook and Hangouts) was split into 60% for training, 20% for validation and 20% for test sets. According to their results, the robustness of the network traffic classifiers is very low in facing ANT and thus there is a reductio in their performance by little random perturbation.
Dias, et al. [14] added an innovative approach for real-time high accuracy video traffic classification based on ML using NB according to the relaxation of the hypothesis of independence. The dataset is video stream containing network layer features. This module represents a solution for network schemes that require adequate real-time traffic treatment. Their results shows that the module is a promising alternative for QoS schemes to be applied in real-time scenarios with good trade of between classification performance and computational costs.
In this paper, a **new network traffic classification model based on NB algorithm has been implemented and tested** and it consists of two main stages and investigate the captured **packet-flow features**. This module can help recognizing different services and applications besides to detecting different protocols into separate classes and construct **the adaptive network based on QoS**.

2.2. Machine Learning Algorithms

Leveraging the set of derived attributes, machine learning techniques were employed to adapt the data and yield classification outcomes [15]. Naïve Bayes (NB) algorithms are guided learning methods that use Bayes' theory combined with the "naïve" assumption of independence between each feature pair [16]. The three types of NB algorithms include: Multinomial NB, Gaussian NB, and Bernoulli NB. Multinominal and Bernoulli NBs will be analyzed here since they are used for NTC in this research.

Multinominal NB (MNB) is a classifier of a C set of classes with N is the size of the dictionary and a test document t_i is assigned to the class of the higher probability $P_r(c|t_i)$ according to the following equation:

$$P_r(c|t_i) = \frac{P_r(c)P_r(t_i|c)}{P_r(t_i)}, c \in C$$

Where $P_r(c)$ represents the prior class, and it is equivalent to the number of items of class $c(N_C)$ divided by the absolute number of archives in the dataset N, $\Pr(c) = \frac{N_c}{N}$. t_i is the document at a given class c. $\Pr(t_i)$ is the class independent input instance probability. $\Pr(t_i|c)$ is the probability of t_i at c and it is calculated as the following equation:

$$\Pr(t_i|c) = \left(\sum_n f_{ni}\right)! \prod_n \frac{\Pr(w_n|c)^{f_{ni}}}{f_{ni}!}$$

where f_{ni} represents the number of words n in t_i, and $\Pr(w_n|c)$ is the probability of word n given class c. This is estimated as:

$$\widehat{\Pr}(w_n|c) = \frac{F_{nc} + 1}{N + \sum_{x=1}^{N} F_{xc}}$$

where F_{xc} represents the quantity of words x in the dataset of c, N represents word reference size. The expression $+1$ at the numerator is utilized to stay away from the zero-frequency issue [17].

The frequency data of every word is caught by the multinominal situation to further develop the characterization results. Greatest deduced assessment [18] is regularly utilized to appraise the boundaries in NB model, including $\Pr(c)$ and $P_r(t_i|c)$. MNB is broadly utilized in text classification issues resulting in one of the most effective algorithms in spam recognition [15], [19], [20].

Bernoulli NB, on the other hand, is a probability distribution is a mathematical function that gives the probabilities of occurrence of different possible outcomes for an experiment

The Naive Bayes theorem provides a conditional probability of an event A occurring provided that an event B has already occurred. The following is its mathematical formula:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Where A and B are two events, $P(B|A)$ is the probability of event A provided event B has already happened and vice versa and finally, $P(A)$ and $P(B)$ are the independent probability of A and B. This theorem can be extended to a list of independent predictors X with one or more class labels Y. In the Bernoulli NB model, features are considered as autonomous binary factors (Booleans) portraying inputs. Assuming x_i is the Boolean variable expressing the event or nonattendance of the $i-th$ word from word reference, after that, the probability of noticing record x given a class c is characterized as follows:

$$P_r(x|c) = \prod_{i=1}^{n} p_c^{x_i}(1 - p_c)^{(1-x_i)}$$

Where, $p_c^{x_i}$ reflects the chance that the term x_i will be found in class c, where c denotes the number of words in the word reference. Because it offers the benefit of showing the absence of terminology, this model has typically been used to describe short texts [15].

Finally, Gaussian NB is another classification algorithm based on defining the likelihood of observing instance t_i for a given class c as described in the following equation:

Technium Vol. 13, pp.12-26 (2023)
ISSN: 2668-778X
www.techniumscience.com

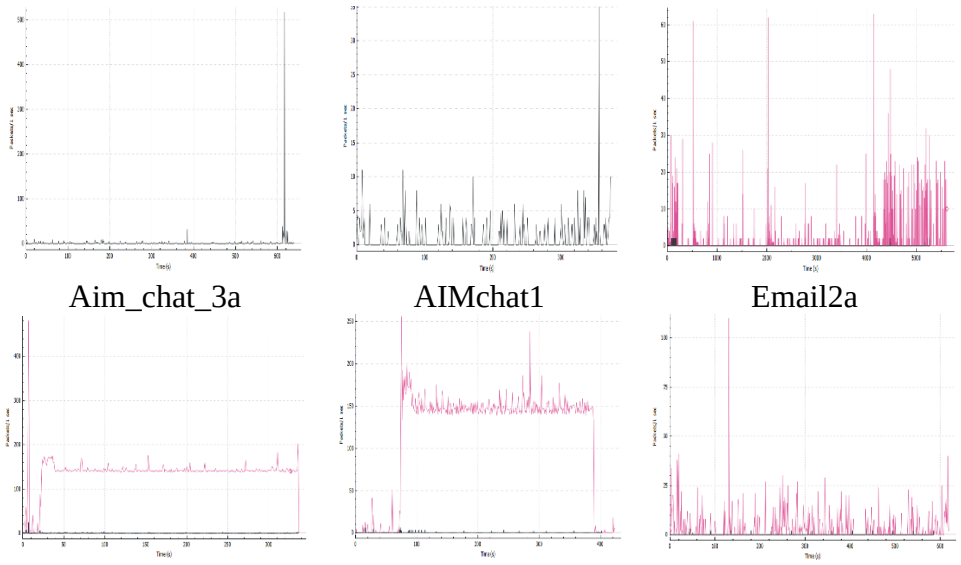
$$P_r(t_i|c) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{(-\frac{(t_i-\mu_y)^2}{2\pi\sigma_y^2})}$$

Where σ_y and μ_y are gaused by maximum likelihood. This algorithm is applied in many prediction examples because of its simplicity and speed [15], [21].

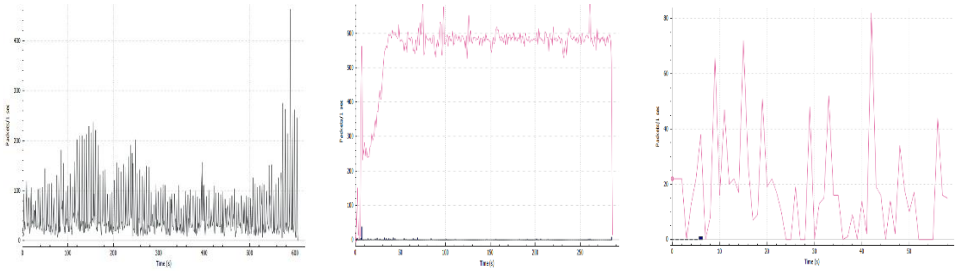
3. CHARACTERISTICS OF SERVICE PROTOCOLS AND DATASET

It is obvious that there are differences between protocols in some services. In this research, two different datasets were used. The first one is the VPN-nonVPN collected dataset (named ISCXVPN2016) which is used for training and testing purposes. It is a real-world network traffic with defined set of tasks to ensure diversity and quantity with virtual users to use different services like Skype, Facebook, etc. for different traffic types (VoIP, P2P, etc.). Regular sessions and a session over VPN were captured, therefore there's an overall of 14 traffic divisions: VOIP, VPN-VOIP, P2P, VPN-P2P, etc. The traffic was grasped with Wireshark and tcpdump, generating an overall quantity of 28GB of information.

ISCXFlowMeter is a Java program that reads pcap files and generates a csv file based on chosen features. The UNB ISCX Network Traffic (VPN-nonVPN) collection comprises of publicly accessible labeled network traffic, including entire packets in pcap and csv formats (flows produced by ISCXFlowMeter) [22]. To determine features, Fig.1 shows the characteristics of trained and tested files which is different among different applications. The second tested dataset is a Wi-Fi session for video browsing stream on the internet on the Google chrome app that is also captured in the pcap file and for 50s. Fig.2 shows characteristics of this dataset.



Technium Vol. 13, pp.12-26 (2023)
ISSN: 2668-778X
www.techniumscience.com



Email1a Facebook_video1a Facebookchat1

Fig.1: Network Traffic Histogram for ISCXVPN2016 dataset

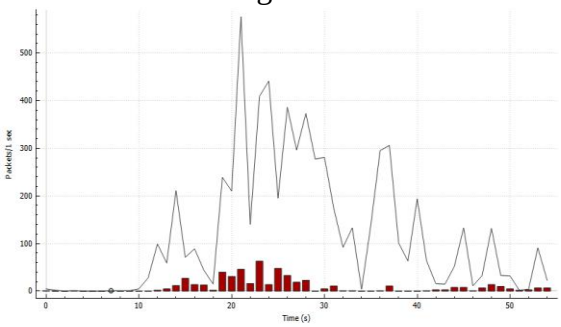


Fig.2: Network Traffic Histogram for Wi-Fi video stream dataset

As a comparison between packet lengths of trained files, it can be seen that video packet lengths ranges from 640 to 1279 packet per second. It is also obvious that QUIC protocol is captured in the WLAN network traffic flow which is quite new protocol used to replace TCP and UDP with adding TLS encryption for establishing (i.e., finishing the connections and transferring data between end-users) connections. Some useful features of this protocol are reduced connection times, better performance when data packets are lost and stable connections when networks are changed.

4. NETWORK TRAFFIC CLASSIFICATION METHOD BASED ON NAVE BAYES

In this paper, a new NTC method has been proposed based on Bernoulli, Gaussian and Multinominal NB technique in ML. This method is based on multi-class classification process to classify the network flows according to networking protocols into the payload data, network security or traffic control. The method flows through two essential steps to get the statistical analysis and results interpretation. The first stage is the preprocessing and feature extraction to build dictionary and generate matrix. The second stage is the application of Bernoulli NB for data fitting and generating predictions.

4.1. Packet flow-based features and pre-processing

In the case of payload, HTTP, UDP, TCP and the new QUIC protocols are responsible for data flow for different applications through the network. Four features are extracted in QUIC-based payloads from client to server and vice versa: the proportion of small packets, the proportion of medium packets, the proportion of large packets, and the average payload length. In the case of security, TLSv1, TLSv1.1, TLSv1.3, SSL, SSLv2, DTLSv1.0 and OCSP protocols are responsible for traffic flow protection against attacks. For secure internet communications, those protocols are typically enabled under the Java Secure Socket Extension (JSSE), which

offers a framework and a solution for a Java version of the TLS and DTLS protocols for data encryption, server authentication, message integrity, and optional client authentication.
The collected dataset is used to capture traffic at the network layer. First, the dataset is preprocessed in the pre-processing phase. This phase includes extracting features of different types of available protocols and building dictionary besides to generating the matrix.

4.2. The proposed method

This section outlines the primary traffic classification for data fitting and classification using Bernoulli and Multinominal NB (see Fig. 3). All protocols will be evaluated as part of the packet flow-based feature extraction in order to create the dictionary and matrix. First, the dataset has been read and specific columns are specified and tested with the *DataFrame-head ()* function. Next, an object with series containing counts of unique is normalized. After that, those matrices will be split into train and test datasets for 67 percent for network training and 33 percent for network testing. The random shuffling was applied on the data before it has been split with an integer passed for reproducible output across multiple function calls. Next, a pipeline of intermediate transforms with a final estimator is applied. The purpose of the pipeline is to collect different phases that can be cross-validated simultaneously during tunning various conditions. Fig.4 explains the pipeline of the Bernoulli NB algorithm. Finally, the sample classification report is created for the statistical analysis.

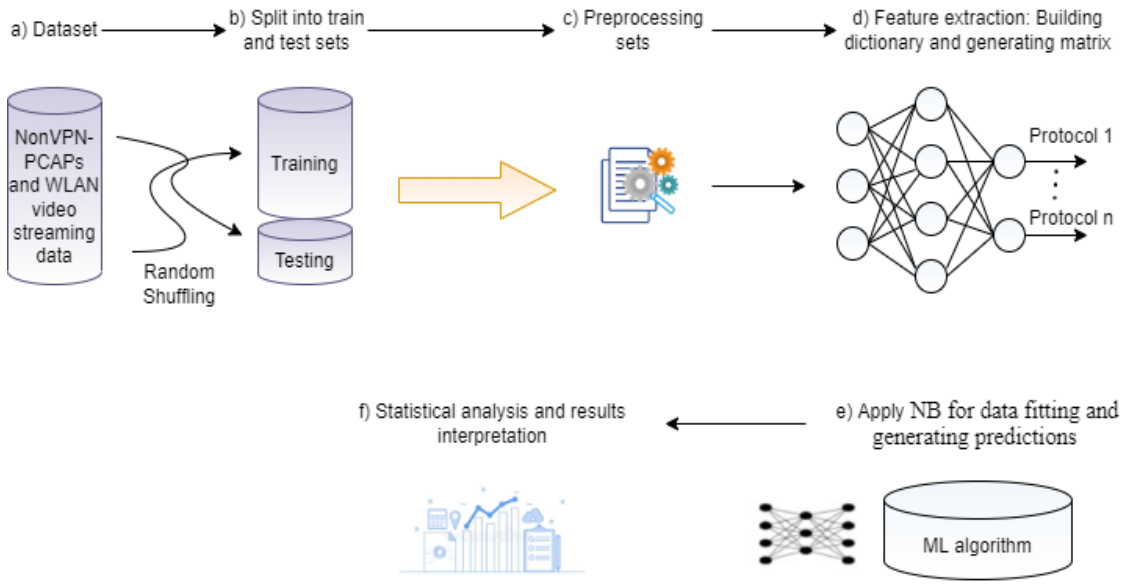


Fig.3: The architecture of the NTC method

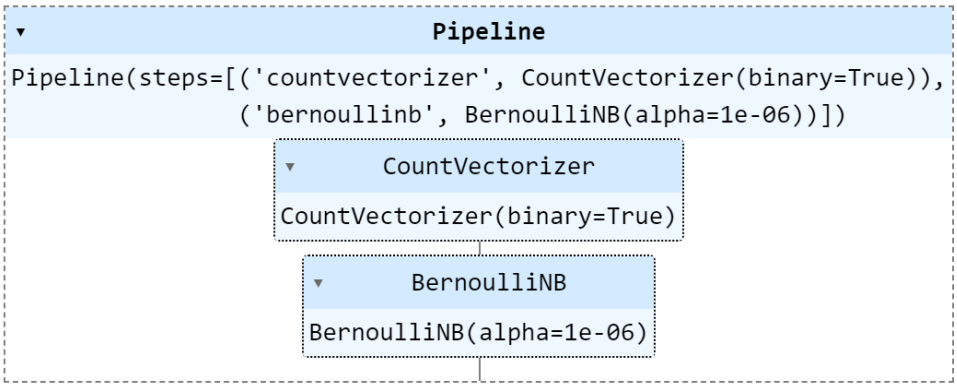


Fig.4: Pipeline of the Bernoulli NB algorithm

5. EXPERIMENTAL RESULTS

5.1. Dataset Specification and Performance Measures

experimentally, a VPN-nonVPN training dataset together with testbed were used to collect network traffic flows. The Wireshark network analyzer on Windows 10 Pro and for 50 seconds was used to record the network flows of online browsing and video streaming. A network traffic of over 172,312 flows with 21 different protocols ranging from security and control protocols to payload protocols. Specification details are described in Table 1. Those traffic flows were divided into 67% for training the model and 33% for testing the model.

Table 1: Dataset Specification

Service	Number of flows
Chat	1,647
Email	34,827
Facebook audio	52,491
Facebook chat	7,719
Facebook video	69,030
YouTube	6,598

The suggested approach was created utilizing the Python Keras library, TensorFlow PowerShell Prompt (tf) environment, Anaconda3 Navigator, and Scikit-Learn tools [23]. In addition, all experiments and implementations were implemented on a Microsoft Surface with processor of Intel(R) Core (TM) i5-6300U CPU @ 2.40GHz 2.50 GHz with installed RAM 4.00 GB and system type 64-bit operating system, x64-based processor running licensed windows 10 pro.

Technium Vol. 13, pp.12-26 (2023)
ISSN: 2668-778X
www.techniumscience.com

Precision, Recall, and F1-score were employed as performance indicators for the suggested method evaluation. Precision is the proportion of the relevant traffic flows that were successfully retrieved. F1-score is the harmonic mean between precision and recall. Equations below represent the details of those parameters [9]:

$$Precision = \frac{TP}{TP + \dot{FP}}$$

$$Recall = \frac{TP}{TP + \dot{FN}}$$

$$F1score = \frac{2}{1/Recall + 1/Precision}$$

Two methods are used to evaluate the overall classification quality [24]. In macro-averaging, a measurement is located in the median value across all similarly treated classes. The combined True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) of all datasets are what determine the micro-averaging. This is called a binary classifier which means characterizing cases as positive or negative. The **positive** implies that the instance is named individual from the class the classifier is attempting to recognize whereas the **negative** means that the instance is delegated as not being an individual from the class the classifier is attempting to distinguish.

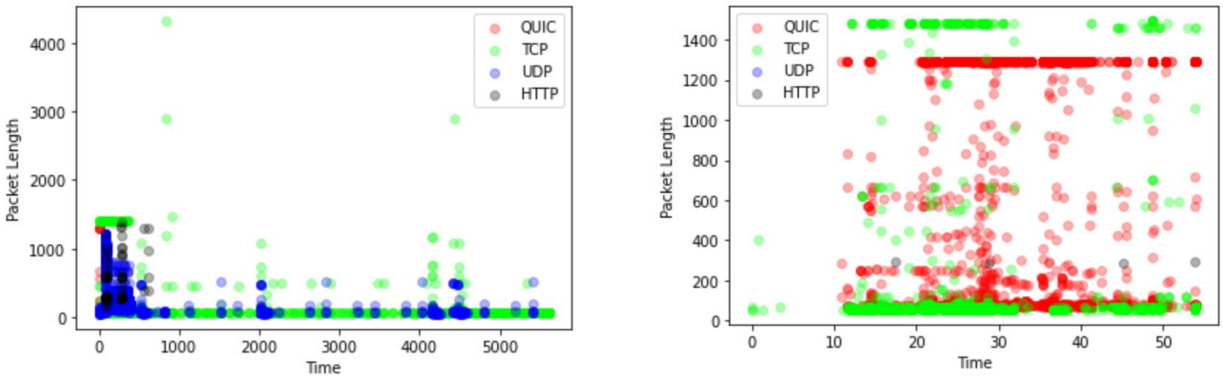
Model evaluation, on the other hand, is often based on the **accuracy** by describing the number of correct predictions over all predictions as shown in the following equation:

$$\frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN} = \frac{No.\ of\ correct\ predictions}{No.of\ all\ predictions}$$
$$= \frac{No.\ of\ correct\ predictions}{N}$$

Where N is the database size. In this research, the **precision** measures how many correct positive predictions (TP). Second, **recall** measures how many positive cases the classifier correctly predicted over all positive cases in the dataset. Finally, **F1-score** measures the average of precision and recall [25].

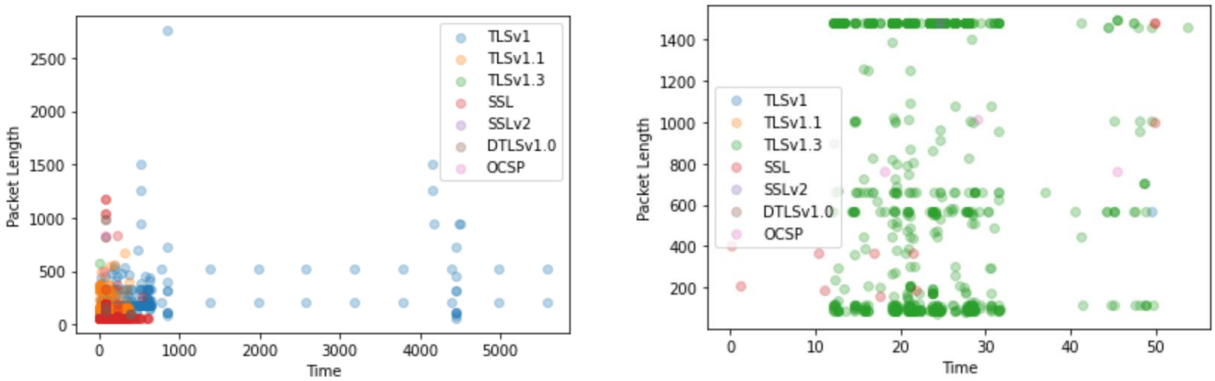
5.2. Experimental Results

In this proposed method, the basic motivation is to get high accuracy on different types of protocols (whether payload or JSSE) for different services and for all three classification methods (Bernoulli, Gaussian or Multinominal). This section is specified to evaluate the performance of the proposed method and highlight advantages over different scenarios including different parts of the dataset.



A) ISCXVPN2016 dataset B) Wi-Fi video stream dataset

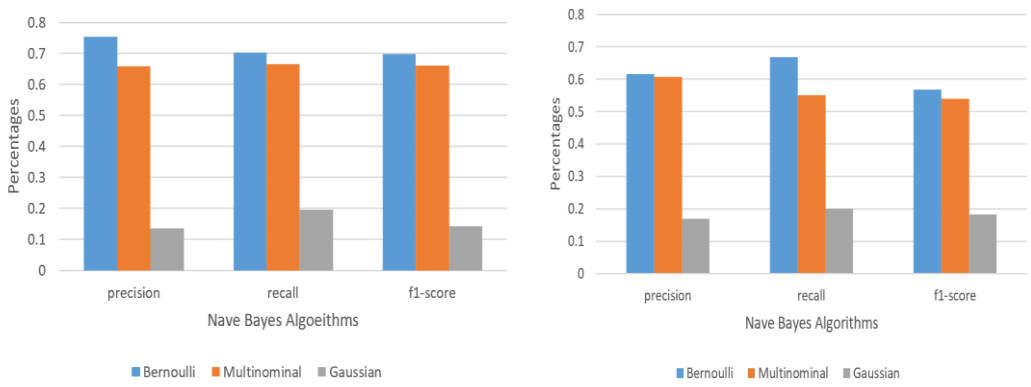
Fig.5: Time vs. Packet Length for payload protocols



A) ISCXVPN2016 dataset B) Wi-Fi video stream dataset

Fig.6: Time vs. Packet Length for JSSE protocols

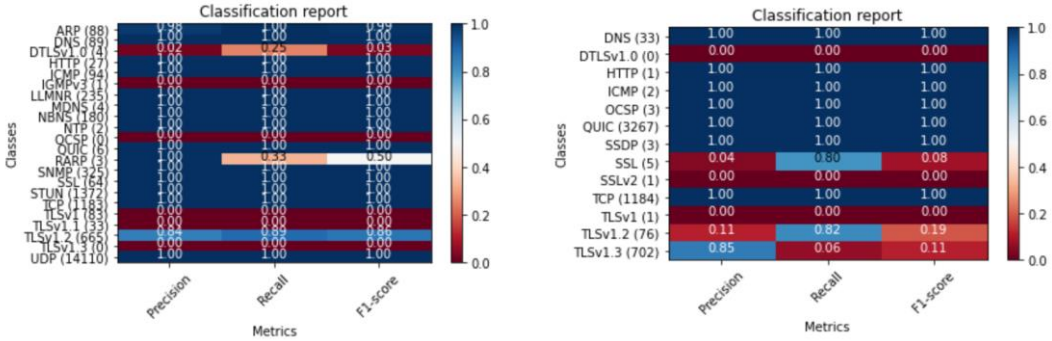
First, datasets were prepared and preprocessed and all packet flow-features were extracted then a Bernoulli pipeline were created. Fig.5 above indicates the distribution of the payload or data transfer protocols for the two datasets according to packet length and time. Fig.6 above, on the other hand, indicates the distribution of the JSSE protocols for both datasets according to packet length and time. Fig.7 (a and b) below, indicate a comparison of the classification report parameters for two different datasets with different protocols (payload, JSSE and other protocols) and for the three algorithms used in this research. Precision, recall, and F1-score in the micro- and macro-averaged datasets are all trending decreasing.



ISCXVPN2016 dataset Wi-Fi video stream dataset

Fig.7: Macro-averaging precision, recall and F1-score.

In the first scenario of the multi-class classification (i.e., for the first dataset), the Bernoulli NB shows an enhanced performance against the other two algorithms in terms of F1-score with accuracy of 98.96% with CPU time of 750 ms whereas the accuracy of Multinominal NB was 99.2% with CPU time of 141 ms and the accuracy of Gaussian NB was 75.98% with CPU time of 75.98 ms. The ratio of payload protocols is 81.97 % while the ratio of JSSE protocols was 4.77 %. In the second scenario (the second dataset), the Bernoulli NB has also the dominant performance but this time in a slight difference with the Multinominal NB in the precision results which in turn affects the F1-score as described in the previous section while for accuracy, the Multinominal NB was higher in 11.21% as compared with Bernoulli NB. The performance was 87.15% with CPU time of 734 ms, 98.36% with CPU time of 15.6 ms and 62.30% with CPU time of 15.6 ms for Bernoulli, Multinominal and Gaussian NBs, respectively. The Bernoulli NB classification report is illustrated in Fig.8 below. The payload ratio for this dataset is 192.82% while JSSE ratio is 17.04%.



A) ISCXVPN2016 dataset B) Wi-Fi video stream dataset

Fig.8: Classification report for Bernoulli NB

6. CONCLUSION

This paper presents an NTC method using ML with TensorFlow and Python programming with NB algorithm which is divided into three types: Bernoulli, Multinominal and Gaussian. This procedure has been implemented and tested on two different datasets for network traffic flows with different protocols. Mainly, important protocols were grouped into two groups: payload

protocols and JSSE or security protocols. Payload protocols include UDP, TCP, QUIC and HTTP while JSSE protocols include SSL, SSLv2, OCSP, DTLSv1.0, TLSv1, TLSv1.2 and TLSv1.3. The two datasets were explained in details earlier in this paper.
From experimental results, we can find that Gaussian classification has a drawback in performance against the other two algorithms although it has shorter CPU time of a 46.85 ms as a ratio compared with 742 ms ratio for Bernoulli and 78.3 ms ratio for Multinominal. There is a tradeoff, on the other hand between Bernoulli and Multinominal in performance and processing time where Bernoulli provides accuracy of 93.05% while the other algorithm provides average accuracy of 98.78% for both datasets (despite of providing 0.6991 rate in F1-score which is slightly higher than 0.662 F1-score rate for multinominal). As a result, we can conclude that the Multinominal classification has more stable performance for different datasets than the other two methods. In the future, a larger dataset will be constructed with deeply investigated full packet Flow-based features to make the proposed method more reliable.

REFERENCES

[1] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey,” *Computer Communications*, vol. 170. Elsevier B.V., pp. 19–41, Mar. 15, 2021. doi: 10.1016/j.comcom.2021.01.021.

[2] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell, “Text Classification from Labeled and Unlabeled Documents using EM,” vol. 39, pp. 103–134, 2000.

[3] A. Shahraki and Ø. Haugen, “An outlier detection method to improve gathered datasets for network behavior analysis in IoT,” *Journal of Communications*, vol. 14, no. 6, pp. 455–462, 2019, doi: 10.12720/jcm.14.6.455-462.

[4] C. Wang, T. Xu, and X. Qin, “Network Traffic Classification with Improved Random Forest,” in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2015, pp. 78–81. doi: 10.1109/CIS.2015.27.

[5] J.-M. Wang, C.-L. Qian, C.-H. Che, and H.-T. He, “Study on Process of Network Traffic Classification Using Machine Learning,” in *2010 Fifth Annual ChinaGrid Conference*, Jul. 2010, pp. 262–266. doi: 10.1109/ChinaGrid.2010.53.

[6] Jomilė Nakutavičiūtė, “What is QUIC protocol used for?,” 2017. Accessed: Jun. 21, 2022. [Online]. Available: https://nordvpn.com/blog/what-is-quic-protocol/

[7] Rohit Dwivedi, “What Is Naive Bayes Algorithm In Machine Learning?,” *Analytic Steps*. https://www.analyticssteps.com/blogs/what-naive-bayes-algorithm-machine-learning (accessed May 28, 2022).

[8] D. Aloraifan, I. Ahmad, and E. Alrashed, “Deep learning based network traffic matrix prediction,” *International Journal of Intelligent Networks*, vol. 2, pp. 46–56, 2021, doi: 10.1016/j.ijin.2021.06.002.

[9] V. Tong, H. A. Tran, S. Souihi, and A. Mellouk, "A novel QUIC traffic Classifier based on Convolutional Neural Networks," *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Sep. 2018, doi: 10.1109/GLOCOM.2018.8647128.

[10] Rui Li, Xi Xiao, Shiguang Ni, Haitao Zheng, and Shutao Xia, "Byte Segment Neural Network for Network Traffic Classification," *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2018, doi: 10.1109/IWQoS.2018.8624128.

[11] L. Chen, J. Liu, and M. Xian, "Network Traffic Classification Using Deep Learning," *International Journal on Artificial Intelligence Tools*, vol. 29, no. 7–8, Dec. 2020, doi: 10.1142/S0218213020400084.

[12] H. K. Lim, J. B. Kim, J. S. Heo, K. Kim, Y. G. Hong, and Y. H. Han, "Packet-based Network Traffic Classification Using Deep Learning," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 046–051. doi: 10.1109/ICAIIC.2019.8669045.

[13] A. M. Sadeghzadeh, S. Shiravi, and R. Jalili, "Adversarial Network Traffic: Towards Evaluating the Robustness of Deep-Learning-Based Network Traffic Classification," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, 2021, doi: 10.1109/TNSM.2021.3052888.

[14] K. L. Dias, M. A. Pongelupe, W. M. Caminhas, and L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, 2019, doi: 10.1016/j.comnet.2019.04.004.

[15] A. Occhipinti, L. Rogers, and C. Angione, "A pipeline and comparative study of 12 machine learning models for text classification," *Expert Syst Appl*, vol. 201, Sep. 2022, doi: 10.1016/j.eswa.2022.117193.

[16] T. M. Mitchell, "Machine Learning and Data Mining," *Commun ACM*, vol. 42, no. 11, pp. 30–36, Nov. 1999, Accessed: Jun. 09, 2022. [Online]. Available: https://www.scopus.com/record/display.uri?eid=2-s2.0-0002337827&origin=inward&txGid=74c09130c9780c4a599def28d44452bb&featureToggles=FEATURE_NEW_DOC_DETAILS_EXPORT:1

[17] A. Mccallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI 1998 Computer Science*, 1998.

[18] Irina Rish, "An Empirical Study of the Naïve Bayes Classifier," *IJCAI 2001 Work Empir Methods Artif Intell*, 2001, [Online]. Available: https://www.researchgate.net/publication/228845263

[19] H. Ney and A. Juan, "Reversing and Smoothing the Multinomial Naive Bayes Text Classifer," in *Pattern Recognition in Information Systems, Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems, PRIS 2002, In*

conjunction with ICEIS 2002, Apr. 2002. [Online]. Available: https://www.researchgate.net/publication/221383148

[20] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," *Springer. Nédellec, C., Rouveirol, C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science*, vol. 1398, 1998, doi: https://doi.org/10.1007/BFb0026666.

[21] R. D. S. Raizada and Y.-S. Lee, "Smoothness without Smoothing: Why Gaussian Naive Bayes Is Not Naive for Multi-Subject Searchlight Studies," *PLoS One*, vol. 8, no. 7, p. 69566, 2013, doi: 10.1371/journal.pone.0069566.

[22] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *ICISSP 2016 - Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, 2016, pp. 407–414. doi: 10.5220/0005740704070414.

[23] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," 2011. [Online]. Available: http://scikit-learn.sourceforge.net.

[24] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/J.IPM.2009.03.002.

[25] Teemu Kanstren, "A Look at Precision, Recall, and F1-Score," *Towards Data Science*, Sep. 12, 2020. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec#:~:text=F1%2Dscore%20when%20Precision%3D0.8%20and%20Recall%20%3D%200.01%20to%201.0&text=Here%20precision%20is%20fixed%20at,varies%20from%200.0%20to%201.0. (accessed Jul. 25, 2022).