Network Traffic Classification with Comparative Analysis Based on KNN vs RF vs ANN

P. PRASANYA DEVI
Guest Lecturer, PG
Department of Computer Science
Government Arts College, Melur
Madurai Kamaraj University
prasanyamsc@gmail.com

SATHIYAPRIYA J
Guest Lecturer, PG
Department of Computer Science
Government Arts College, Melur
Madurai Kamaraj University
skannanmku@gmail.com

***Abstract*---The importance of network traffic classification has grown over the last decade. Coupled with advances in software and theory, the range of classification techniques has also increased. Network operators can predict demands in future traffic to high accuracy and better identify anomalous behavior. Multiple machine learning tools have been developed in this field and each have had varying degrees of success. Network traffic classification is the operation of giving appropriate identification to every traffic flow through a network. Several methods have been applied in the past, to achieve network traffic classification including port-based, payload- based, behavior based and so on. These methods have been found some limitation or the other. Nowadays, attention is now on Machine Learning (ML) methods that rely on the statistical properties of the traffic flows generated. However, ML methods do not perform well when confronted with large-scale traffic data having large number of features and instances.. In this study, network traffic classification using ML methods is demonstrated from two perspectives: one that involves feature selection and one that does not. A number of performance metrics are considered including runtime, accuracy, recall, precision and F- score. . This paper presents the design and implementation of , a flow-based network traffic classifier for online applications. and uses three ML models, namely K-Nearest Neighbors (KNN), Random Forest (RF), and Artificial Neural Network (ANN), for classifying the most popular online applications, including Amazon, Youtube, Google, Twitter, and manyothers.**

Keywords---*Machine Learning, Deep Learning, Network Traffic Analysis, KNN, RF, ANN*

I. INTRODUCTION

Network traffic analysis is the process of recognizing user applications, networking protocols, and communication patterns flowing through the network [1]. Traffic analysis is useful for identifying security threats, intrusion detection, server performance deterioration, configuration errors, and latency problems in some network components [2]. The rapid evolutions of new online applications, as well as the ubiquitous deployment of mobile and IoT devices [3], have dramatically increased the complexity and diversity of network traffic analysis. moreover, the new security requirements in modern networks, including packet encryption and port obfuscation, have elevated extra challenges in classifying network traffic [4].

Despite the importance, the traditional network traffic classification approaches can only recognize user applications that are running over static well-known network ports such as FTP, SSH, HTTP, SMTP, etc. However, most online user applications use dynamic ports, virtual private networks, and encrypted tunnels [5]. Furthermore, these applications are transported over HTTPS connections and have applied security protocols (e.g., SSH and SSL) for ensuring QoS provisioning, security, and privacy. This makes it very challenging for traditional port-based approaches to recognize such applications.

Several techniques have been applied to classify network traffic data, the most common being port-based technique. The port-based technique makes use of the port number assigned to applications by the Internet Assigned Number Authority (IANA) . The limitation of this technique comes from the use of dynamic port numbers instead of the originally assigned or well-known port numbers. Also, applications can hide under another application such that their own port number is not captured but that of the host application [4]. The second known technique is the payload-based technique. This technique also known as deep packet inspection (DPI) works by inspecting the entire payload of each packet to discover the signature pattern of the packet.

The DPI method has the issue of packet encryption which makes it difficult to detect some applications and /or their correct attributes during the classification process . The technique also makes it possible for third parties who have no connection with the traffic to inspect the payload of each packet . Due to the limitations of previous techniques attention is now toward ML techniques that make use of statistical properties of the traffic flows. However, ML techniques do not perform efficiently when confronted with dataset that possess non-relevant and redundant features. Table 1 presents a summary of the network traffic classification methods. The table details the characteristics and limitations of the different classification methods.

TABLE 1
OVERVIEW OF NETWORK TRAFFIC CLASSIFICATION METHODS

S/N	Classification Method	Characteristics/Advantages	Disadvantages
1	Port-Based [5][8]	• Traffic identification done using port numbers allocated by IANA. • Method is fast and low-resourceconsuming • Supported by many network devices	• Due to growing number of application, there is tendency to use unpredictable port numbers • The method may not be suitable for applications not registered by IANA • Method may not work well with applications that use dynamically allocated port numbers.
2	Deep Packet Inspection Method [5][9]	• Inspects the actual pay-load of thepacket. • Identification not based on portnumber • Method provides more accurate result compared to port-based techniques • Method is quite suitable for P2Ptraffic	• Method is slow and requires much processing power- high computational cost • Signatures must be kept up-to-date, as the applications change very frequently • Not easy to apply to encryptedtraffic. • The method also suffers from violation of privacy policies and regulations
3	Machine Learning Method based Statistical Analysis of attributes [5][6][7]	• Method is based on analysis of statistical properties of the flow comprising the packets. • Attributes of flow such as packet size, packet inter-arrival times etcmay be used. • Method is fast and consumes less processing power • It can also detect the class of yet unknown applications • Method is also able to identify encrypted traffic.	• There is usually the need to preprocess the traffic data before classification • The performance of the ML algorithms may be affected by too many features especially when they either not relevant or redundant. Feature selection is usually employed to select optimal featuresets.

II. LITERATURE SURVEY

Hassan Alizadeh,et.al(2020) suggested the innovative method in way to classed the network traffic with an implemented GMM (Gaussian Mixture Model) [11]. The Component wise expectation maximization abbreviated as CEM was exploited for making a separate GMM in way so that the traffic distribution was go with similarity. The suggested had classified and verified the traffic on time efficiently using only preliminary packets of truncated flows. A publicly available dataset taken from a real network was utilized for conducting the experiments in order to compute this technique. The experimental outcomes demonstrated that the suggested technique had attained the accuracy around 97.7% for classifying the network flow in comparison with other methods.

Won-Ju Eom, et.al (2021) introduced a model recognized as LightGBM model with the help of SDN (software- defined network) architecture for classifying the network traffic [12]. This model was established in the network controller with the purpose of leveraging the better computational capacity of the SDN controller to classify the network traffic in real-time, adaptively and accurately. Four ensemble algorithms were deployed and their efficacy to classify the model was analyzed. Moreover, the suggested model performed more effectively in classifying the network traffic.

Madhusoodhana Chari S.,et.al(2019) intended the packet size signature extracting based method for the classification distinct classes including Audio and Video streaming ,the Browsing, Chat, P2P etc. [13]. For this purpose, the classes of network traffic were recognized by the training a J48 DT (decision tree) classification algorithm with a new feature set. The interpretability of the model was described. This set had provided a tree which was found more balanced and capable of producing the lower count of rules for individual class. The set provided interpretability to the intended method and easy deployment.

Jiwon Yang, et.al (2019) projected a traffic classification technique to classify the encrypted traffic flows [15]. A new payload-based classification was put forward using which the unencrypted handshake packets were utilized that had exchanged amid the hosts to stable the transport layer security. The Bayesian neural network was employed as the classed technique where the cipher suite, compression technique related to the suite-packets was considered as the insertion. The investigation were carried out and outcomes depicted that the projected technique performed more efficiently in comparison with other conventional payload-based classification algorithms. The future work would focus on extending by classifying other secure protocols.

Yu Wu, et.al (2018) designed an approach for enhancing the classic, The time-division multiplexing Ethernet passive optical network framework [16]. Meanwhile designed approach made the deployment of two methods. Initially, the ML (machine-learning) models were deployed in order to classify the upstream traffic as useful and useless classes.

Subsequently, sifting useless traffic was applied for avoiding the transmission of redundant EPON (Ethernet passive optical network) frames. The optimal outcomes were obtained by integrating baseness of 2 classifiers in the integrated method using 2 feature- selection techniques. In the second method, the hybrid bandwidth allocation system had utilized it as an input. The simulation outcomes revealed that the designed algorithm offered promising improvements with regard to per-RRH traffic load and SNR, the expanded form is signal-to-noise ratio and kept the E2E, the expanded form is end-to-end delay under 100 μs.

Xinxin Tong,et.al(2020) suggested an innovative classification called Bidirectional Flow Sequence Network on the basis of the long short-term memory [20]. Different from the conventional classifier, the BFSN was an E2E (end-to-end) classifier assisted in learning the representative attributes from the traffic and classification. Furthermore, the bidirectional traffic succession developed by the usage of the length and direction knowledge of encrypted traffic, processed this algorithm on the basis of LSTM. The ISCX VPN dataset was utilized for conducting experiments. And the experimental output depicted the suggested classifier yielded accuracy up to 91%.

III. ANALYSIS OF KNN VS RF VS ANN

A. **The Random Forest Algorithm**- As the name suggests, a Random Forest is a tree-based ensemble with each tree depending on a collection of random variables. More formally,for a p-dimensional random vector ,

X = (X1,...,Xp) T representing the real-valued input or predictor variables and a random variable Y representing the real-valued response, we assume an unknown joint distribution PXY (X,Y). The goal is to find a prediction function f(X) for predicting Y. The prediction function is determined by a loss function L(Y, f(X)) and defined to minimize the expected value of the loss

EXY (L(Y, f(X))) (1)

where the subscripts denote expectation with respect to the joint distribution of X and Y. Intuitively, L(Y, f(X)) is a measure of how close f(X) is to Y; it penalizes values of f(X) that are a long way from Y.

Typical choices of L are squared error loss L(Y, f(X)) = (Y − f(X))^2 for regression and zero-one loss for classification:

L(Y, f(X)) = I(Y 6= f(X)) = 0 if Y = f(X) 1 otherwise. (2)

It turns out (see, for example, [10] Sect. 2.4) that minimizing EXY (L(Y, f(X))) for squared error loss gives the conditional expectation

f(x) = E(Y|X = x) (3)

otherwise known as the regression function. In the classification situation, if the set of possible values of Y is denoted by Y , minimizing EXY (L(Y, f(X))) for zero-one loss gives f(x) = argmax y∈Y P(Y = y|X = x), (4) otherwise known as the Bayes rule.

Ensembles construct f in terms of a collection of so-called "base learners" h1(x),...,hJ (x) and these base learners are combined to give the "ensemble predictor" f(x). In regression, the base learners are averaged

f(x) = 1 J J ∑ j=1 hj(x), (5)

while in classification, f(x) is the most frequently predicted class ("voting")

f(x) = argmax y∈Y J ∑ j=1 I(y = hj(x)).(6)

In Random Forests the jth base learner is a tree denoted hj(X,Θj), where Θj is a collection of random variables and the j's are independent for j = 1,...,J.

B. **ANN** (Artificial Neural Network) -ANN (Artificial Neural Network) can be referred to as both the natural and artificial alternatives, though classically this term is used to refer to artificial and external systems only. Mathematically, neural nets are regarded as nonlinear objects with each layer representing non-linear combination/variations of non-linear functions from the prior layers. Each neuron in the network is a multiple-input, multiple-output (MIMO) scheme that receives pointers from the inputs, produces a subsequent signal, and communicates that signal to all possible outputs. ANN has algorithms in different forms to help address the problem domains that Artificial Neural Network (ANN) solves. Some of the algorithms include:

a) Back propagation with single modified neuron
b) Back propagation with linear neuron
c) Matrix approach
d) Gradient Descent.

The back propagation algorithm has been the most popular approach for neural networks training/classification due to its flexibility and robustness. This method has been used to solve used to solve various real life problems. The network is a multiple-input, multiple-output (MIMO) scheme that receives pointers from the inputs, produces a subsequent signal, and communicates that signal to all possible outputs.

Basically, neurons in an Artificial Neural Network (ANN) are arranged into different discrete layers. The first and topmost layer is the one that interacts with the surroundings to receive various combinations of possible input is known as the input layer. The last and final layer that interacts with the output to present the final processed data is known as the output layer

While the layers that are between the input and the output layer that do not have any real communication with the environment are known as hidden layers. Hence increasing the complexity of an Artificial Neural Network (ANN), and its computational ability, requires the additions of a lot of more hidden layers and neurons per layer

C.*K-Nearest Neighbors* *(K-NN)* -K-NN classifies an instance by considering the majority of the surrounding instances. Whatever class or label that majority of the neighbors of a particular instance belong, that is the class assigned to that instance[11].

Feature Selection Methods Used

Five popular ranker-based feature selection algorithms were used for this study. Ranker-based feature selection methods are computationally efficient and are based on different

metrics thereby suitable for making some comparisons.

Information Gain (IG)

IG is a very common univariate filter technique. It evaluates features based on the information they have gained from other features. IG first classifies all the features, and then following a threshold, a certain number of features are selected based on the order obtained [12].

ReliefF

ReliefF is a multivariate ranking-based method that works by arbitrarily sampling an instance and then finding its nearest neighbor from the same and opposite class. The idea is that a useful feature should be able to separate between samples that belong to different classes. that the KNN model is not adequate to be used for real-time inference. In contrast, on average, the ANN model requires around 250 ms only to detect a security thread in a traffic flow. To put these numbers in some context, consider a firewall service installed in a gateway router; our ANN model can inspect around four applications' flow streams per second, which would have a negligible latency overhead over the network stream

IV. RESULT

Table 2. defines the performance analysis of Machine learning Table 3compares the KNN, RF, and ANN models in terms of classification accuracy and prediction time across the 53 classes. For instance, the ANN model achieved an overall average classification accuracy of 99.16%. The average prediction time of the model was measured to be 0.25 s. This is evident that administrators can detect any security vulnerability in their networks using a handy web-based GUI in a quarter of a second. Furthermore, we noted that the prediction accuracy of many classes (e.g., Netflix, Amazon,etc.) was 99.6%. This shows that our model is robust and can operate in real-time inference in real-world network settings with high accuracy. Figure 1shows the precision, recall and F1-score of the RF model-chart, Figure 2. Classification Accuracy(%)Compared to ANN and RF, KNN for different classes. Figure 3. Prediction Time(Seconds) Compared to ANN and RF, KNN for different classes. Compared to ANN and RF, KNN is

TABLE 3.
THE PRECISION, RECALL AND F1-SCORE OF RF

Class	Precision	Recall	F1-Score
AMAZON	0.95	0.95	0.95
FACEBOOK	0.93	0.94	0.94
FTP_CONTROL	0.78	0.78	0.78
FTP_DATA	0.99	1	0.99
GMAIL	0.84	0.84	0.84
GOOGLE	0.95	0.99	0.97
GOOGLE_MAPS	0.97	0.97	0.97
HTTP	1	1	1

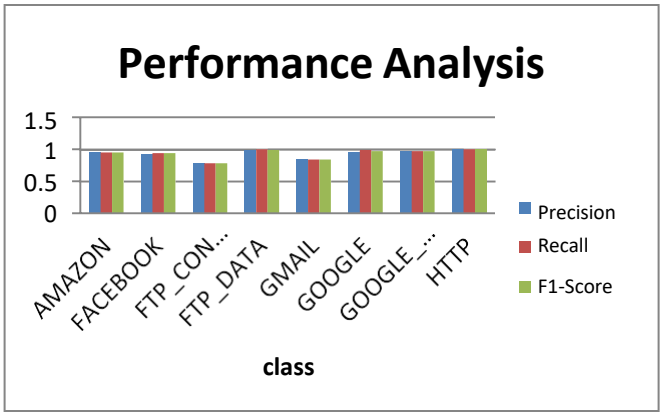

Fig 1.The Precision, Recall and F1-Score of the RF Model-Chart

TABLE 2.
THE AVERAGE CLASSIFICATION ACCURACY % AND PREDICTION TIME (SECONDS) OF KNN VS.RF VS ANN

	Prediction Time			Prediction Time		
Class	**KNN**	**RF**	**ANN**	**KNN**	**RF**	**ANN**
AMAZON	283	0.74	0.29	283	0.74	0.29
FACEBOOK	197	0.65	0.4	197	0.65	0.4
FTP_CONTROL	144	0.01	0.09	144	0.01	0.09
FTP_DATA	171	0.66	0.34	171	0.66	0.34
GMAIL	193	1.01	0.28	193	1.01	0.28
GOOGLE	306	0.69	0.29	306	0.69	0.29
GOOGLE_MAPS	159	0.76	0.27	159	0.76	0.27
HTTP	334	0.65	0.31	334	0.65	0.31
NETFLIX	141	0.88	0.27	141	0.88	0.27
NTP	307	0.54	0.27	307	0.54	0.27
OFFICE_365	173	0.76	0.28	173	0.76	0.28
SKYPE	139	0.77	0.28	139	0.77	0.28
SPOTIFY	152	0.75	0.28	152	0.75	0.28

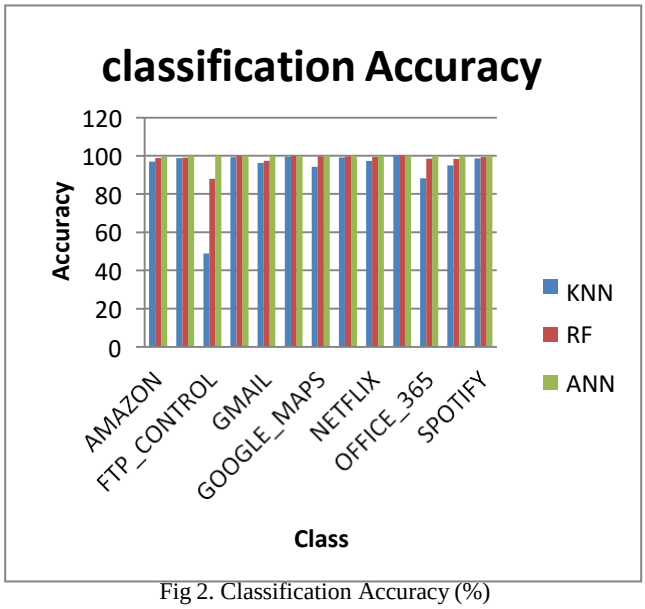

Fig 2. Classification Accuracy (%)

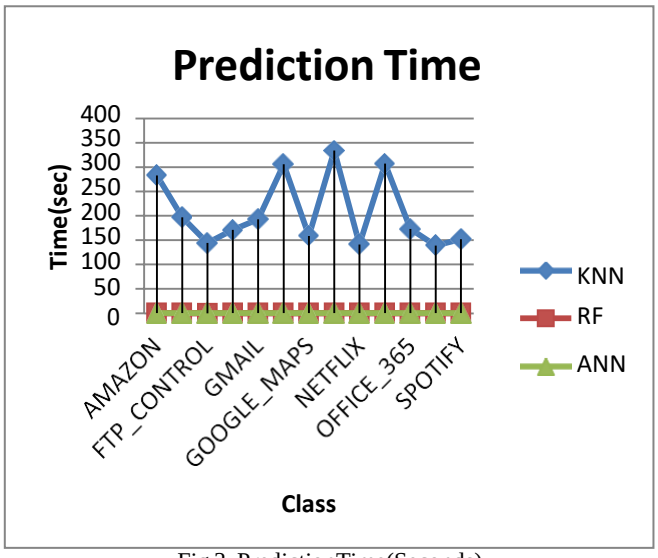

Fig 3. PredictionTime(Seconds)

V. CONCLUSION

The system implementation compared three ML models, namely ANN, RF, and KNN, in terms of classification accuracy and performance. To increase the system usability, we developed a user-friendly interface on top of these models to allow users to interact with the system conveniently. We carried out several sets of experiments for evaluating the performance and classification accuracy of our system, paying particular attention to the prediction time. Our ANN model could most notably inspect four applications' flow streams per second, which proves that is suitable for real-time inference at the edge with offline generated ML models.

REFERENCES

[1] Rezaei, S.; Liu, X. Multitask learning for network traffic classification. In Proceedings of the International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 3–6 August 2020; pp. 1–9.

[2] Lotfollahi, M.; Zade, R.S.H.; Siavoshani, M.J.; Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Comput. Springer Link 2020, 24, 1999–2012. [CrossRef]

[3] Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access 2017, 5, 42–50. [CrossRef]

[4] Moamen, A.M.A.; Hamza, H.S. On securing atomic operations in multicast aodv. Ad-Hoc Sens. Wirel. Netw. 2015, 28, 137–159.

[5] Zeng, Y.; Gu, H.; Wei, W.; Guo, Y. Deep-Full-Range: A deep learning based network encrypted traffic classification and intrusion detection framework. IEEE Access 2019, 7, 182–190. [CrossRef]

[6] Hassan Alizadeh, Harald Vranken, André Zúquete, Ali Miri, "Timely Classification and Verification of Network Traffic Using Gaussian Mixture Models", 2020, IEEE Access

[7] Mendiola, J. Astorga, E. Jacob, M. Higuero, "A survey on the contributions of Software-Defined Networking to Traffic Engineering," IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 918 -953, 2017.

[8] M. Karakus, A. Durresi, "Quality of Service (QoS) in Software Defined Networking (SDN): A survey," Journal of Network and Computer Applications, vol. 80, pp. 200-218, 2016.

[9] M. Dusi, R. Bifulco, F. Gringoli, F. Schneider, "Reactive logic in software-defined networking: Measuring flow-table requirements," IEEE International Conference on Wireless Communications and Mobile Computing (IWCMC), pp. 340-345, 2014.

[10] F. Hu, Q. Hao, K. Bao, "A survey on software-defined network and openflow: From concept to implementation," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2181-2206, 2014.

[11] H. Kim, N. Feamster, N, "Improving network management with software defined networking," IEEE Communications Magazine, vol. 51, no. 2, pp. 114-119, 2013.

[12] M. R. Parsaei, S. H. Khalilian, R. Javidan, "A Comparative Study on Fault Tolerance Methods in IP Networks versus Software Defined Networks," International Academic Journal of Science and Engineering. Vol. 3, no. 4, pp. 146-154, 2016.

[13] T. J. Parvat, P. Chandra, "A Novel approach to deep packet inspection for intrusion detection," Procedia Computer Science, vol. 45, pp. 506-513, 2015.

[14] N. Williams, S. Zander, G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," ACM SIGCOMM Computer Communication Review, vol. 36, no. 5, pp. 5-16, 2006.

[15] H. Cui, Y. Zhu, Y. Yao, L. Yufeng, Y. Liu, "Design of intelligent capabilities in SDN," IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), pp. 1-5, 2014.