Deep Learning Approach for Real-time Video Streaming Traffic Classification

1st Mohammed Al Jameel
Department of Computing and Immersive Technologies
University of Northampton
Northampton, United Kingdom
mohammed.aljameel@northampton.ac.uk

2nd Scott Turner
School of Computing
Canterbury Christ Church University
Canterbury, United Kingdom
scott.turner@canterbury.ac.uk

3rd Triantafyllos Kanakis
Department of Computing and Immersive Technologies
University of Northampton
Northampton, United Kingdom
triantafyllos.kanakis@northampton.ac.uk

4th Ali Al-Sherbaz
Department of Technical and Applied Computing
University of Gloucestershire
The Park, Cheltenham, United Kingdom
aalsherbaz@glos.ac.uk

5th Wesam S. Bhaya
College of Information Technology
University of Babylon
Babil, Iraq
wesambhaya@uobabylon.edu.iq

***Abstract*—Video streaming services such as Amazon Prime Video, Netflix and YouTube, continue to be of enormous demands in everyday peoples' lives. This enticed research in new mechanisms to provide a clear image of network usage and ensure better Quality of Service (QoS) for these applications. This paper proposes an accurate video streaming traffic classification model based on deep learning (DL). We first collected a set of video traffic data from a real network. Then, data was pre-processed to select the desired features for video traffic classification. Based on the performance evaluation, the model produces an overall accuracy of 99.3% when classifying video streaming traffic using a multi-layer feedforward neural network. This paper also evaluates the DL approach's effectiveness compared to the Gaussian Naive Bayes algorithm (GNB), one of the most well-known machine learning techniques used in Internet traffic classification. The model is promising to be applied in a real-time scenario as it showed its ability to predict new unseen data with 98.4% overall accuracy.**

***Keywords*—Traffic classification, Video streaming, Deep learning, Multi-layer feedforward neural network**

I. Introduction

Video streaming services continue to be of tremendous demand. Recently, a rapid spread of video streaming applications over the internet has been observed, and according to the 2018 Cisco Visual Networking Index (VNI) [1] IP video traffic will be 82% of all consumer Internet traffic by 2022, up from 75% in 2017 while, Ultra High Definition (UHD) IP video will account for 22% of global IP video traffic. With a focus on meeting the traffic requirements of such huge capacities, there will be an absolute necessity to efficiently utilise the available network resources such as bandwidth demands with the application requirements. This is essential to ISPs in which they will be encouraged to consider different methods to provide better QoS for their clients [2]. It is crucial to classify and identify various network applications to understand network conditions, which forms a framework for managing networks such as load balancing, bandwidth allocation and route optimisation.

Clearly, video streaming services such as YouTube, Netflix or Amazon Prime are commonly known as *bandwidth-hungry* services in modern network [3], which are source of challenges to the Internet Service Providers (ISPs) as they can be influenced by delay, packet loss, jitter and bandwidth limitations. Such impairments affect the quality of the video streaming which may result in a poor QoS, hence a poor Quality of Experience (QoE) [4] [5]. Classification and identification of video streaming traffic are key to bandwidth allocation for the aggregated traffic flows from clients and ensures better QoS of different applications [6] [7].

Flow-based traffic classification has recently gained the attention of the research community as it overcomes the limitations of the traditional methods of network traffic classification in a *supervised* or *unsupervised* [8] manner. In the network traffic classification, in case of whether it refers to a particular application or not, the classification can be either *coarse-grained* or *fine-grained* classifications [9]. The first one performs identification and classification of the entire network traffic whereas the second one, as shown in the rest of this paper, refers to the fine classification of specific application range.

This paper employs a multi-layer feedforward neural network algorithm to present an effective real-time video streaming traffic classification model. The model classifies three video streaming services (Amazon Prime, Netflix and YouTube) as a solution of bandwidth allocation, improvement of QoS and QoE, and network optimisation [10]. Effective feature extraction and processing methods are researched and adapted to achieve a classification accuracy of 99.3%. The model produces excellent classification decisions on new captured features with 98.4% overall accuracy.

II. Characterisation and Classification of Video Streaming

Video streaming applications can be used for entertainment, security, or self-diagnosis. Video streaming will require more bandwidth due to the demand for higher image quality [11]. In order to meet the traffic requirements of huge files such as UHD or 4K video, the available resources should be utilised efficiently. A 2011 research [12] investigated the network features of YouTube and Netflix. It showed that the influence of streaming strategy is very important as it fluctuates on the applications (web applications, mobile applications) and the container/protocols (Flash, HTML5, Silverlight). Throughout a session of normal streaming, video traffic is transmitted in two phases: a buffering phase succeeded by a steady-state phase as it is shown in Fig. 1. There is an on-off cycles periods appeared in the steady-state stage which employed in order to limit the download rate.

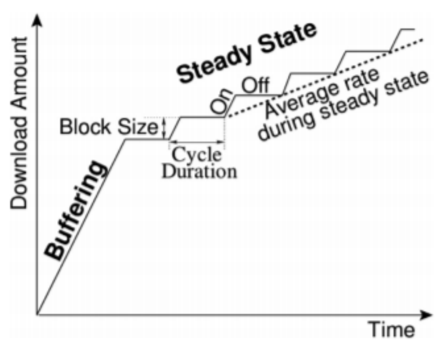

Fig. 1: Generic behaviour of a video streaming [12].

The dominant approaches in traffic classification inspects the communication ports in the TCP/UDP header and linked them with well-known ports to decide which applications produced the traffic [13]. The modern video transmission is over ports 80 and 443 (http and https respectively) as this secures the transmission of information over web applications. Machine Learning based techniques are implemented to classify traffic flows without requiring deep inspection of packet contents [14]. Without regard to the customised algorithms, researchers are looking at these techniques as the best substitute because they have a much lower computational cost and are able to detect encrypted traffic [15].

Concerning classification of video traffic, several research applied the traditional machine learning (ML) methods in their approaches. Authors in [16] proposed uplink/downlink rate as traffic classification features. They adopted support vector machine as their classifier. The experiments result proved that the proposed mechanism reached an accuracy of 98.98%. Another work proposed by Bakhshi and Ghita [17] considered YouTube, Netflix and Dailymotion as the target streaming services. They used a two-phased ML classification mechanism in their approach. K-means was used to group the traffic classes, and a decision tree to classify the applications in order to provide more granularity to their results.

Dong, Zhao and Jin [18] defined a scheme to classify internet video traffic. They considered a flow of 5-tuple in their approach and calculated more than 40 statistical features from this tuple, categorising them into upstream and downstream. After the adaptation of information gain ratio, based on consistency-based feature selection filtering; four out of twelve features were selected. The experimental resulted classifier accuracy reached more than 98% for the six tested types of video applications.

Researchers in [2] focused on individual classification of video streaming. They adopted relaxation of the hypothesis of independence between attributes of the naive Bayes algorithm to increase the accuracy of traffic classification. Their experiment considered YouTube, Netflix and file download. Upon the extraction of 14 features, correlation graph was applied as the selection technique. They evaluated their approach against the classic Gaussian Naive Bayes with an out-preforming accuracy of 98.88% over the 85.25% traditional approach.

A simple multi-layer perceptron neural network based on Markov Decision Process proposed in [19] to classify five streaming video services YouTube, YouTube TV, Netflix, Amazon Prime, or HBO. In the training stage, 23 features were used as input, a single hidden layer with 4 nodes, and a ReLU transfer function was applied in the hidden layer. The classification results showed that the highest accuracy was occurred when classifying Netflix data with a 92% while YouTube TV traffic obtained the lowest accuracy result which reached to 84.5%.

The work in [20] defined a method to classify network traffic flows by using principal component analysis (PCA) technique together with six ML algorithms. They paid attention to the pre-processing phase as they adopted the 20 features presented in [21], and used the feature of server responding duration in their classification experiment.

Ling-Yun et al. [22] studied and analysed the features of video flow during the transmission process and statistical features of its main protocol. They introduced new features based on video downlink rate probability distribution and UDP/TCP packet number. Correlation-based feature algorithm was employed as a feature selection technique and traditional ML techniques were used to identify video streaming.

From this research, it is safe to claim that most studies paid attention to identifying network applications based on categories rather than looking at the fine-grained classification of specific application scope with an exception to [2] [19]. This paper's approach focuses on the individual classification of video streaming traffic with the help of DL techniques.

Practical features extraction and processing methods are researched and adapted to achieve a DL model with a classification accuracy of 99.3%. The DL model also takes an excellent classification decision on new unseen traffic with 98.4% overall accuracy.

III. Video streaming traffic classification model

Essential steps to classify the video streaming traffic include the collection of a real data, data processing which involves multiple processing phases to obtain an optimal subset of features to be used in the classification experiment and data training. Fig. 2, shows the architecture of the proposed model:

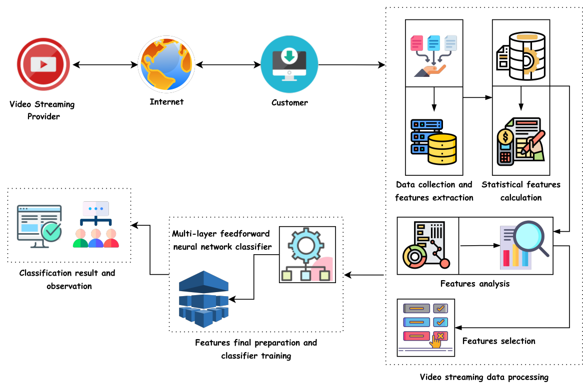

Fig. 2: The proposed architecture of deep learning model.

A. Data Acquisition

An appropriate dataset was needed for the research, a decisive experiment was considered to generate online data from multimedia streaming services (Amazon Prime Video, Netflix and YouTube). With the use of Wireshark as traffic monitoring software on those services, data traffic was captured in four intervals with four minutes per capture resulting in a total time of sixteen minutes per vendor, this then was exported in a JSON formatted file. Due to the fact that supervised learning was used, traffic identification needed undertaking. Following that, was the extraction of data from the IPv4 header fields from the resulted JSON files.

Table. I displays the name, type, and descriptions of the extracted variables from the JSON file. Afterwards, all data was recorded in different files based on the source that it was extracted from, with every service's data in a single file, Table. II.

B. Statistical Features Computation

The data was operated in streaming settings, meaning that some raw attributes might not refer to the streaming characteristics. Therefore, we applied a statistical features calculation of the video streaming traffic in a window with the desired size. Before statistics generation, it is noteworthy to mention that we captured video stream traffic in four intervals, and one of the captured features was $frame\_time$. $frame\_time$ indicates the date and time of the arrival of packets. A method of $frame\_time$ processing was proposed in this paper and

TABLE I: Extracted features from JSON file.

Name	Type	Description
frame_time	Date and time	Arrival Time, which indicates date and time. The format is: MMM dd, yyyy hh:mm:ss.SSSSSS
frame_len	Unsigned integer	Frame length
frame_number	Unsigned integer	Frame number
frame_cap	Unsigned integer	Frame length (capture length)
eth_dst	MAC address	Destination mac address
eth_src	MAC address	Source mac address
ip_len	Unsigned integer	Total length
ip_frag_offset	Unsigned integer	IP fragment offset
ip_ttl	Unsigned integer	IP time to live
ip_proto	Unsigned integer	IP protocol
ip_src	IPv4 address	Source ip address
ip_dst	IPv4 address	Destinatio ip address
UDP_tcp_srcport	Unsigned integer	UDP, TCP Source Port
UDP_tcp_dstport	Unsigned integer	UDP, TCP Destination Port
tcp_ack	Unsigned integer	TCP acknowledgment number
tcp_window_size	Unsigned integer	Calculated window size
UDP	Integer	To determine if the protocol used is TCP or UDP. It holds a value of 1 if TCP occurs and 0 when UDP occurs.
UDP_tcp_len	Unsigned integer	UDP, TCP Length

TABLE II: Dataset detail for the three types of video streaming traffic.

Streaming type	Number of packets	Class
Amazon Prime	133182	1
Netflix	69999	2
YouTube	750755	3

applied after finishing the data collection step. The concept behind this was to generalise each capture regardless of the exact time of video traffic capturing experiments. At the beginning of capture, time was initially set to zero; following this, the arrival time of each packet was taken in hours, minutes, seconds and milliseconds. This helps considering the difference of milliseconds between packets in all captures. In addition, the $frame\_time$ was also processed before the creation of windows. In this sense, the first packet of the window was set to zero. The following packets of the window was also calculated based on the first arrival time of a packet. Accordingly, all windows were handled in the same behaviours.

The window size is flexible and can be changed to meet the experiment requirements. A window of three packets was considered. For instance, an experiment involving a window of two packets showed that the result was biased to the sharpest value in the window. On the other side, in the case with a higher value, i.e., five packets, despite reducing the amount of data by five times, the result was not that different from the

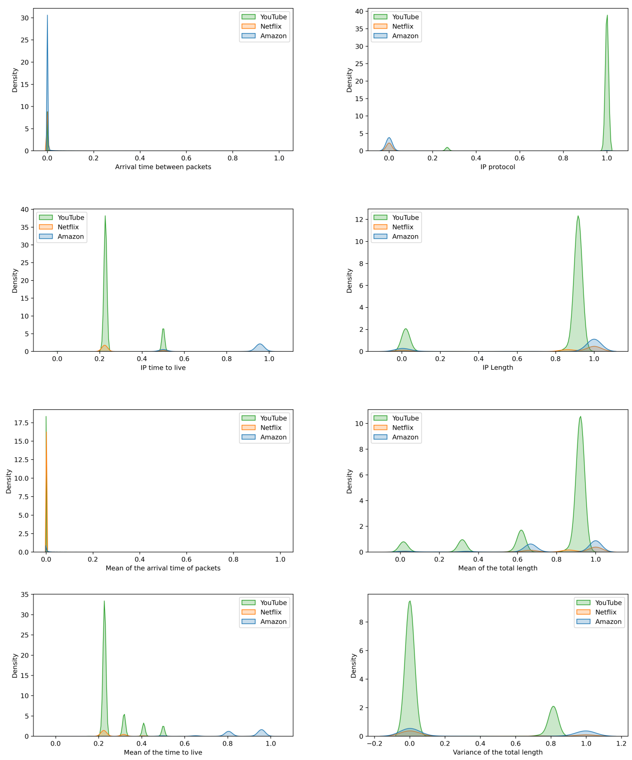

Fig. 3: Graphs of features with their density distribution.

window of three packets. A script was developed to calculate each window's mean, variance, median, standard deviation, min and max. The outcome of this script was 38 statistical features that can be used to classify the video streaming traffic. Following the creation of the windows, a method was developed to handle labelling each packet based on the source applications as presented in Table. III. The overall windows generated was 317976 windows of packets.

C. Features Analysis

Following the statistical features computation step, observation of the density graph of the examined non-statistical and statistical features was needed to define their distribution. The reason behind this was to highlight the notion that statistics need to be computed and features are beneficial to be used for the classification experiment. Each feature in the density graphs is normalised, and the value is mapped to the interval [0, 1] and describes the density of the data distribution. Fig. 3 illustrates the density distribution of some features used in the classification experiment. The density graphs indicate that $ATBP$ and $MATOP$ features are not evenly distributed in their interval and display narrow peaks. Additionally, it is observed that ip_proto, ip_len and $VIPL$ show bimodal shapes. Other values stated a multimodal distribution, such as $MTTL$ and $MIPL$.

TABLE III: Video streaming statistical features computed using a window of three packets.

Feature name	**Description**
ATBP	Arrival time between packets
MATOP	Mean of the arrival time of packets
VATOP	Variance of the arrival time of packets
SATOP	Standard deviation of the arrival time of packets
MNATOP	Minimum value of the arrival time of packets
MXATOP	Maximum value of the arrival time of packets
MDFL	Median of the frame length
MNFL	Minimum value of the frame length
MXFL	Maximum value of the frame length
SFL	Standard deviation of the frame length
MDIPL	Median of the total length (IP datagram length)
MIPL	Mean of the total length
VIPL	Variance of the total length
SIPL	Standard deviation of the total length
MNIPL	Minimum value of the total length
MXIPL	Maximum value of the total length
MDPL	Median of the protocol length (UDP, TCP length)
MPL	Mean of the protocol length
VPL	Variance of the protocol length
SPL	Standard deviation of the protocol length
MNPL	Minimum value of the protocol length
MXPL	Maximum value of the protocol length
MDOF	Median of the IP fragment offset
SOF	Standard of the IP fragment offset
MNOF	Minimum value of the IP fragment offset
MXOF	Maximum value of the IP fragment offset
MDPIP	Median of the protocol value in the IP datagram
MPIP	Mean of the protocol value in the IP datagram
VPIP	Variance of the protocol value in the IP datagram
SPIP	Standard deviation of the protocol value in the IP datagram
MNPIP	Minimum value of the protocol value in the IP datagram
MXPIP	Maximum value of the protocol value in the IP datagramm
MDTTL	Median of the time to live value in the IP datagram
MTTL	Mean of the time to live value in the IP datagram
VTTL	Variance of the time to live value in the IP datagram
STTL	Standard deviation of the time to live value in the IP datagram
MNTTL	Minimum value of the time to live value in the IP datagram
MXTTL	Maximum value of the time to live value in the IP datagram
CLASS	Classification of each packet

D. Features Selection

Before the selection of features, it is noteworthy that the data presented in numeric values with different scales and had different units therefore, the data was standardised and normalised for each of the extracted features [23]. Following this, Pearson's Product-Moment Correlation Coefficient matrix is used to find out the strength and direction of the relationship of variables. It can be defined as one of the well-known measures of correlation [24]. The idea of applying this matrix was to observe feature-to-feature association. The values of this matrix are between -1 and 1, and those values indicate the strength of the correlation [25]. The value of Correlation coefficient is given by equation. 1:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (1)$$

Where n, corresponds to the quantity of information, while x and y corresponds to the first and second variable values, respectively.

The output of this procedure is shown in Fig. 4. It is worth noting that $MDOF$, SOF, $MNOF$, $MXOF$ and $MNATOP$ have not been included when applying the feature selection method as their values are equal to zero.

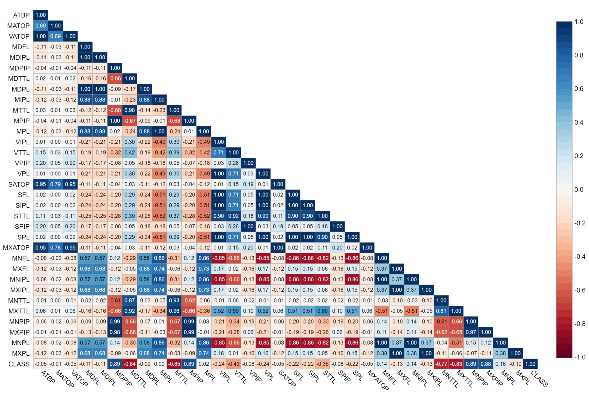

Fig. 4: Pearson's product-moment correlation coefficient matrix of the statistical features.

Based on the graph observation, it can be noticed that some variable pairs have correlations equal to one, which indicates a perfect correlation for the pair. In this scenario, we eliminate one variable of every pair from the base that have a correlation higher than 0.9, for the sake of keeping the highly independent ones. For example, we excluded one feature of each of the pairs ($ATBP$, $VATOP$) and (VPL, $VIPL$) from the classification experiment as their correlation values are equal to one. The pair ($MDTTL$, $MTTL$) have a high correlation however, the removal of such features is not recommended. The notion behind this was to benefit the performance of the proposed DL model. In conclusion, ten features ($ATBP$, $MATOP$, $MDFL$, $MDPIP$, $MDTTL$, $MIPL$, $MTTL$, $VIPL$, $VTTLL$, $VPIP$) were selected to be used in the training experiment. The dataset, as well as all the scripts used, are available in [26]

E. Video Streaming Data Training

The DL model used in the proposed approach was a multi-layer feedforward neural network. It is a biologically inspired classification method. It comprises of a number of simple neuron-like processing nodes, arranged in layers. Each node in a layer is connected with all the nodes in the previous layer [27]. It is important to mention that the feedforward neural network uses a supervised method called back-propagation. It propagates backwards from the output layer to the input in order to decrease errors and enhance performance by modifying all the connection weights. Levenberg-Marquardt backpropagation algorithm is used to train the proposed network. If the aforementioned procedure performed once for each pattern and class pair in the dataset, it means that one epoch of learning has been completed [28]. The classification experiment applied in MATLAB (R2020b) installed in ACER Aspire V 15 Nitro laptop.

The activation functions between the input, hidden, and output layers decide whether the network model can converge faster, indicating accurate prediction. Non-linear activation functions are usually performed to better fit and improve compatibility [29]. Therefore, a log-sigmoid transfer function was applied in the hidden layer, and a tan-sigmoid transfer function was applied in the output layer to train the data. The structure was built using the ten features acquired by applying Pearson's Product-moment correlation coefficient matrix as the input layer and the three multimedia types as targets. Based on several experiments, the number of neurons in the hidden layer was set to twenty, fifteen and ten neurons in each of the hidden layers, respectively, as shown in Fig. 5. In both experiments, the proposed DL model was compared with the GNB model and evaluated based on the result of the confusion matrix.

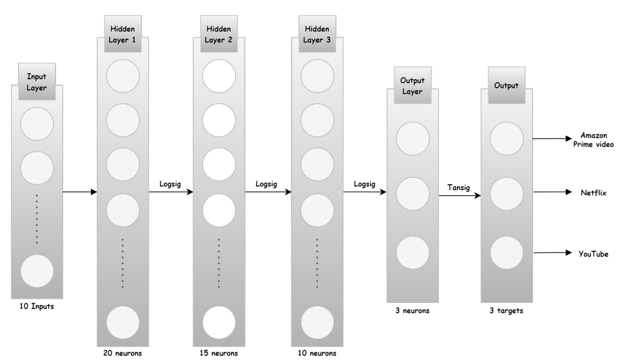

Fig. 5: Overview design of the multi-layer feedforward neural network.

The following equations are usually required to measure the quality of network traffic classifier [30].

$$Accuracy = AC = \frac{\sum_{i=1}^{T} \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{T} \quad (2)$$

where T corresponds to the total number of classes, TP_i and TN_i refer to the true positive and true negative for i class, and FP_i and FN_i indicate the false positive and false negative for a particular class.

$$Precision = Prec_i = \frac{\sum_{i=1}^{T} \frac{TP_i}{TP_i + FP_i}}{T} \quad (3)$$

$$Recall = Rec_i = \frac{\sum_{i=1}^{T} \frac{TP_i}{TP_i + FN_i}}{T} \quad (4)$$

The following parts illustrate the outcomes which contain the result of these matrices provided from the confusion matrix.

IV. Results and discussions

A. First Experiment

The first experiment involved the use of GNB model. The result of confusion matrix is shown in Table. IV. The overall accuracy produced from this model achieves classification of 97%, meaning that it obtained an error rate of 3%.

It also shows a very low false positive rate in YouTube class reaches to 0.2% and an acceptable one for Amazon Prime data reaches 6.5%, which means that samples were classified as other classes (Netflix or YouTube) without being so. Unlike both classes, the model achieves a very low precision for the second class which represents Netflix data, reaches to 78.1% since 21.9% of the class samples classified as other classes while they are not. Furthermore, the model presents a low rate of the class one samples that were not classified as such, which reaches to 1.8%. This results in a high recall when classifying elements of such class which reaches to 98.2%. Similarly, for Netflix and YouTube samples, the sensitivity of both groups reaches to 94% and 97.1% respectively.

TABLE IV: Classification accuracy of both models with 10 features.

Num. of class	Class type	GNB model		DL model	
		Prec.	*Rec.*	*Prec.*	*Rec.*
3	Amazon Prime	93.5	98.2	97.5	98.4
	Netflix	78.1	94.0	97.9	95.3
	Youtube	99.8	97.1	99.7	99.8
Overall accuracy		97.0		99.3	

The dataset has been applied in the proposed multi-layer feedforward neural network. The validation and testing data in this model were randomly divided using a random data division function in 80% for training using a Levenberg-Marquardt backpropagation algorithm, 10% for validation and 10% for testing. The overall accuracy produced by the algorithm achieves classification of 99.3%, meaning that it obtained a low error rate of 0.7%.

The evaluation results of DL model are shown in Table. IV. The model achieves high precision for all applications, which reaches to 97.5%, 97.9% for both Amazon Prime and Netflix streaming services. Specifically, YouTube data approaches 99.7%, since only 0.3% of the class samples classified as other classes without being so. This means that the DL model presents a low false positive rate in all the three classes. Similarly, recall is high for all applications, which reaches an average result of 97.8%. In comparison with the GNB model, multi-layer feed forward neural network has relatively high performance. In fact, the results of recall and precision are much better when comparing with the GNB model. This leads to a high accuracy of 99.3%.

B. Second Experiment

The DL model aims to classify traffic in real-time. With that in mind, an experiment took place to implement real traffic from the services in the trained model. A one-computed window is enough to identify the service due to the fact that a window of three packets is considered to obtain the statistical features. The window can be achieved during ten milliseconds of capturing real-time data traffic. 90 seconds samples of each streaming service were obtained to achieve the real-time classification of multimedia traffic. The experiment went through the same stages when collecting and processing the data in the training experiment. It is noteworthy that this data has not been experimented with.

In the first scenario, the GNB model has been applied. Based on result displayed in Table. V, almost the three classes have been classified correctly, which leads to an accuracy of 96.9%. Despite that the data used in this scenario was untrained, a fresh set, the classifier model presented a good classification result for the new set. The table also shows that the model has a little high false positive rate of class one (Amazon Prime) which reaches to 19.3% that leads to 81.7% precision. However, the model performed well for other groups and showed a low false positive rate for class two and three, thus leading to a precision of 96.7% and 99.9%, respectively. Moreover, the classifier presents a low false negative rate in all classes, which leads to an average result of 96.8% recall for all classes.

While in the second scenario, the DL model has achieved a very high accuracy result which reaches to 98.4%. In comparison with the GNB model, in Table. V, apart from the YouTube data, this approach has a relatively higher precision when classifying Amazon Prime and Netflix data than in the GNB model which leads to 89.2% and 99.3%, respectively. However, it shows a slightly lower precision compared to the GNB model reaches to 99.7%. Additionally, the low false negative rate in all application is even lower than in the GNB model, which results in high recall of 98.7%, 95.3% and 99.4%, for each class respectively.

TABLE V: Classification accuracy of both models with 10 unseen features.

Num. of class	Class type	GNB model		DL model	
		Prec.	*Rec.*	*Prec.*	*Rec.*
3	Amazon Prime	81.7	98.3	89.2	98.7
	Netflix	96.7	94.5	99.3	95.3
	Youtube	99.9	97.5	99.7	99.4
Overall accuracy		96.9		98.4	

V. Conclusion

The classification of individual video streaming is essential solution for efficient network resource management and ensuring better QoS in line with each application requirements. The proposed approach of DL model was able to classify Amazon Prime, Netflix, and YouTube streaming videos with an overall accuracy of 99.3%. To experience a real-time scenario, an experiment was carried out to collect a new dataset from the aforementioned streaming services. The new data was treated and fed to the trained DL model. The experimental result based on the unseen data showed that almost the three classes have been classified correctly, which leads to an accuracy of 98.4%.

The demonstrated result also approved that the proposed DL model is inspiring and promising to be applied in real-time scenarios. In future research, we will continue to improve the proposed approach and apply the same concept in SDN network. The idea behind that was to optimise the network performance by introducing an automated bandwidth allocation method for video streaming traffic. With the help of DL and the promising features of SDN, the model can classify the traffic in real-time and allocate bandwidth for the aggregated traffic flows from clients.

References

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (vni), complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018.

[2] 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, pp. 143–157, 2019.

[3] A. Ellis and M. Sorokina, *Optical Communication Systems: Limits and Possibilities*. CRC Press, 2019.

[4] X. Huang, T. Yuan, G. Qiao, and Y. Ren, "Deep reinforcement learning for multimedia traffic control in software defined networking," *IEEE Network*, vol. 32, no. 6, pp. 35–41, 2018.

[5] J. Frnda, M. Voznak, and L. Sevcik, "Impact of packet loss and delay variation on the quality of real-time video streaming," *Telecommunication Systems*, vol. 62, no. 2, pp. 265–275, 2016.

[6] N. Carlsson, D. Eager, V. Krishnamoorthi, and T. Polishchuk, "Optimized adaptive streaming of multi-video stream bundles," *IEEE transactions on multimedia*, vol. 19, no. 7, pp. 1637–1653, 2017.

[7] P. Tang, Y. Dong, J. Jin, and S. Mao, "Fine-grained classification of internet video traffic from qos perspective using fractal spectrum," *IEEE Transactions on Multimedia*, 2019.

[8] F. Audah, T. S. Chin, R. Kapsin, N. Omar, and A. Tajuddin, "Future direction of traffic classification in sdn from current patents point-of-view," in *2019 15th International Computer Engineering Conference (ICENCO)*. IEEE, 2019, pp. 121–125.

[9] E. Biersack, C. Callegari, M. Matijasevic *et al.*, "Data traffic monitoring and analysis," *Lecture Notes in Computer Science*, vol. 5, no. 23, pp. 12 561–12 570, 2013.

[10] A. Canovas, J. M. Jimenez, O. Romero, and J. Lloret, "Multimedia data flow traffic classification using intelligent models based on traffic patterns," *IEEE Network*, vol. 32, no. 6, pp. 100–107, 2018.

[11] "Cisco Annual Internet Report (2018–2023)," *Cisco*, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[12] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, 2011, pp. 1–12.

[13] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," 1998.

[14] L. AlSuwaidan, "Data management model for internet of everything," in *International Conference on Mobile Web and Intelligent Information Systems*. Springer, 2019, pp. 331–341.

[15] H. A. H. Ibrahim, O. R. A. Al Zuobi, M. A. Al-Namari, G. MohamedAli, and A. A. A. Abdalla, "Internet traffic classification using machine learning approach: Datasets validation issues," in *2016 Conference of Basic Sciences and Engineering Studies (SGCAC)*. IEEE, 2016, pp. 158–166.

[16] W. Zai-jian, Y.-n. Dong, H.-x. Shi, Y. Lingyun, and T. Pingping, "Internet video traffic classification using qos features," in *2016 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2016, pp. 1–5.

[17] T. Bakhshi and B. Ghita, "On internet traffic classification: A two-phased machine learning approach," *Journal of Computer Networks and Communications*, vol. 2016, 2016.

[18] Y.-n. Dong, J.-j. Zhao, and J. Jin, "Novel feature selection and classification of internet video traffic based on a hierarchical scheme," *Computer Networks*, vol. 119, pp. 102–111, 2017.

[19] A. Shaout and B. Crispin, "Streaming video classification using machine learning," *The International Arab Journal of Information Technology*, vol. 17, no. 4A, pp. 677–682, 2020.

[20] Y. Miao, Z. Ruan, L. Pan, J. Zhang, and Y. Xiang, "Comprehensive analysis of network traffic data," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 5, p. e4181, 2018.

[21] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Transactions on Parallel and Distributed systems*, vol. 24, no. 1, pp. 104–117, 2012.

[22] L.-Y. Yang, Y.-N. Dong, W. Tian, and Z.-J. Wang, "The study of new features for video traffic classification," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 15 839–15 859, 2019.

[23] L.-H. Chang, T.-H. Lee, H.-C. Chu, and C. Su, "Application-based online traffic classification with deep learning models on sdn networks," *Adv. Technol. Innov*, vol. 5, pp. 216–229, 2020.

[24] R. Rendall, I. Castillo, A. Schmidt, S.-T. Chin, L. H. Chiang, and M. Reis, "Wide spectrum feature selection (wise) for regression model building," *Computers & Chemical Engineering*, vol. 121, pp. 99–110, 2019.

[25] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[26] M. Al Jameel, "Deep learning approach for real-time video streaming traffic classification," 2021. [Online]. Available: https://github.com/mo7ammedfadhil/Video-streaming-dataset

[27] N. Namdev, S. Agrawal, and S. Silkari, "Recent advancement in machine learning based internet traffic classification," *Procedia Computer Science*, vol. 60, pp. 784–791, 2015.

[28] M. R. Parsaei, M. J. Sobouti, S. R. Khayami, and R. Javidan, "Network traffic classification using machine learning techniques over software defined networks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, pp. 220–225, 2017.

[29] A. Wuraola and N. Patel, "Sqnl: A new computationally efficient activation function," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[30] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.