Int. j. inf. tecnol. (September 2020) 12(3):805–824
https://doi.org/10.1007/s41870-020-00458-0

ORIGINAL RESEARCH

Performance analysis of network traffic capture tools and machine learning algorithms for the classification of applications, states and anomalies

T. P. Fowdur[1] · B. N. Baulum[1] · Y. Beeharry[1]

Received: 13 July 2019 / Accepted: 11 April 2020 / Published online: 23 April 2020

Abstract Network analytics is of key importance for the proper management of network resources as the rate of Internet traffic continues to rise. The aim of this paper is to investigate the performance of different network traffic capture tools for extracting features and to evaluate the performance of eight Machine Learning (ML) algorithms in the classification of (1) applications; (2) states and (3) anomalies. Six Internet applications were considered along with four PC states and two network anomalies. The network was monitored by three traffic capture tools: PRTG, Colasoft Capsa and Wireshark and classification was performed using the Weka Toolkit. The performance of the eight ML classifiers was determined based on several metrics. The Colasoft Capsa feature set gave the highest accuracy for the classification of applications while same was achieved with features from PRTG for the classification of the four states considered. For anomaly classification, the ML algorithms showed almost similar classification behavior when the Colasoft Capsa or PRTG feature set was used.

Keywords Network analytics · Traffic monitoring · Machine learning · Applications · Anomalies · Attacks

✉ T. P. Fowdur
p.fowdur@uom.ac.mu

B. N. Baulum
bibi.baulum1@umail.uom.ac.mu

Y. Beeharry
y.beeharry@uom.ac.mu

[1] Department of Electrical and Electronics Engineering, University of Mauritius, Reduit, Mauritius

1 Introduction

Network traffic, most commonly referred to as the amount of data being transferred across a network at a specific time, is increasing at a drastic rate as the Internet continues to grow in scope and complexity [1]. Network traffic can also be measured in terms of bandwidth or transmission capacity and is an important factor when determining the quality and speed of a network. The emergence of more and more applications running on Internet Protocol (IP) networks in different fields—including not only traditional Internet services such as WWW, FTP, and e-mail, but also multimedia services such as multimedia streaming, P2P file sharing and gaming—has yielded to network bandwidth growing from hundreds of Mbps to busier and faster wireless networks of more than 10 Gbps [2]. It is therefore crucial for networks to be monitored so as to understand their behavior in terms of applications and bandwidth usage, utilization of network resources, and to detect network anomalies and security issues, hence preventing overall network performance degradation or failure. The two main operations encompassing network analytics are traffic monitoring and traffic classification. Network traffic monitoring tools are employed by administrators in order to check for availability and maintain system stability by fixing network problems on time and ensuring the network security strength. On the other hand, traffic classification helps to identify different applications and protocols that utilize the network's resources. While network analytics is not essential for private networks, it is an indispensable tool for large business operators to have a better understanding of their networks and which eventually enables them to make smarter and data-driven decisions to attain desired operations' outcomes and to meet customers' needs. In other words, the process involves the study of

network data and statistics to identify trends and patterns for easy detection and elimination of anomalies [3]. An overview of recent publications that have proposed interesting classification approaches of IP traffic is given next.

In [4], Parsaei et al. applied ML algorithms on captured traffic from a Software-Defined Network (SDN). Four ML algorithms, namely feedforward, Multi-layer Perceptron (MLP), the Levenberg–Marquardt and Naïve Bayes were used. To specify specific flows, features like source port, destination port, IP source, IP destination and transport layer protocol were used. Testing of the classifier model yielded to an accuracy of 95.6% for feedforward, 97% for MLP and Levenberg–Marquardt and finally 97.6% for Naïve Bayes algorithm. The study successfully attained its objective of minimizing overhead of controllers' processing and network traffic. In [5], a comparative analysis of ML algorithms for classification of traffic from internet applications was performed. For data collection, real time network traffic for a duration of one minute using Wireshark software was collected and the Weka toolkit was used for classification. Traffic from WWW, DNS, FTP, P2P and Telnet applications were targeted. The classification model was constructed by the application of four machine learning algorithm, namely Naive Bayes, Bayes Net, C4.5 and Support Vector Machine (SVM). It was found that C4.5 algorithm gave the highest classification accuracy at 79%. The results also revealed that the recall and precision values for DNS and WWW applications are lower than those of the remaining applications.

In [6], Singh and Agrawal conducted a classification of IP traffic using ML approach. The performance of the five ML algorithms was evaluated based on parameters such as classification accuracy, training time and precision and recall values. It was found that for the case of full feature dataset, the Bayes net classifier gave the best classification accuracy, which is 85.3%. A 100% recall and precision value was recorded with Bayes Net for FTP, P2P, VoIP and IM. In [7], Sohi et al. made use of three ML algorithms: Bayes Net, RBF and C4.5 for classifying Internet traffic into educational and non-education applications. Some educational websites used were the IEEE, Science Direct and SparkNotes while non-educational sites included BitTorrent and Yahoo Messenger. It was found that Bayes Net gave a classification accuracy of 76.6%, making it the most accurate among the 3 classifiers. The latter also outperformed the RBF and C4.5 classifiers in terms of recall and precision for both educational and non-educational Internet applications.

In [8], the authors presented several criteria to assess existing network data capture mechanisms. An extensive review of state of the art network data collection techniques such as packet, flow and log based methods was performed with an in depth analysis of their benefits and drawbacks using the proposed criteria as a means for systematic evaluation. The evaluation criteria used system performance indicators such as instantaneity, effectiveness, scalability and expense among others as a basis. A number of open problems were also identified and several possibilities for future research were identified. In [9], a study based on the selection of features from network traffic in the detection of anomalies was made. The work focused on data preprocessing and outlined the importance of feature selection. This step helps to remove redundant features and hence allows for faster processing and storing of data by reducing resource consumption. To evaluate the performance of the selected feature set, ML algorithms such as KNN, Naïve Bayes, Decision Trees, Artificial Neural networks (ANN) and SVM were deployed. They assessed the performance of the classifiers with datasets consisting of 41, 30 and 16 features. It was observed that the classifiers performed better with feature sets of smaller size. The Bayes classifier showed a high False Positive rate by considered almost every new sample as attack with 41 features. However, its performance greatly improved with 16 features, but at the cost of less anomaly detection power.

Building upon the works previously described, this paper aims at analysing the network traffic of an 802.11 wireless LAN by first capturing a maximum amount of traffic information from on-going sessions of internet applications using three network monitoring tools, namely PRTG, Wireshark and Colasoft Capsa. The applications employed are YouTube, Skype, BitTorrent, Google Drive, Browsing and FTP sharing. Traffic generated during downloading, uploading, streaming and idle states are also captured. The collected data are then used in the evaluation of 8 ML classification algorithms, serving as analytic tools. Moreover, the effect of anomalies in the form of DDoS attack and rogue servers on the network performance is also examined.

The remainder of this paper proceeds as follows: Sect. 2 describes how each traffic capture tool is used for feature extraction. Section 3 describes the classification algorithms used for the analytics and how to perform the analytics with the Weka Toolkit. Section 4 describes the system model used for capturing and analyzing network traffic for different applications, states and anomalies. Section 5 presents the results of all extracted features and classification results of each scheme as well as evaluation and analysis on the performance of each classifier with different feature sets. Section 6 concludes the paper with some recommendations for future works.

2 Feature extraction tools

Based on previous researches, three network monitoring tools were chosen for conducting this study. They are PRTG, Wireshark and Capsa. Their main features are outlined in the following subsections.

2.1 PRTG

PRTG [10, 11] is a product of Paessler which serves as a network monitoring tool. While PRTG is not capable of functioning as an intrusion detection system, it acts as a preventive system and warns against anomalous activities in a network.

Key features of PRTG:

- Monitoring of network performance in terms of bandwidth and application usage.
- Monitoring of system usage (CPU loads, free memory, free disk space) of hardware devices.
- Makes use of a statistical approach by setting up threshold values for traffic parameters and hence detects and alerts about anomalies like unexpected load peaks and abnormally heavy traffic, downtimes and slow servers. Spikes in activity can signal a threat.
- User-friendly graphics engine that makes network activity accessible in the form of tables and graphs and hence facilitates analysis of network usage.
- Efficient database system that provide storage of raw monitoring data and a report generator to create both live and scheduled reports in CSV, HTML or XML data files.
- Network analysis modules for automatic discovery of network devices and sensors.

Several sensors are used by PRTG to track and display network traffic. Four sensors have been deployed for traffic capture and feature extraction. They are the Windows Network Card, Ping [12], DNS and health sensors.

2.2 Wireshark

Wireshark [13, 14] is an open-source network protocol analyser or sniffer that captures and displays data traversing a network in the form of packets. The main features of Wireshark include:

- Ability to perform live capture of packets and deep offline analysis of protocols and packet contents.
- Reading of live data from several interfaces such as IEEE 802.11, Ethernet, Bluetooth, ATM, USB, among others.
- Provide powerful filters for selecting specific protocols for analysis.
- Use of coloring rules to highlight packets for quick and easy identification of different protocols.

The captured traffic obtained from Internet applications is saved as CSV files for further processing.

2.3 Colasoft Capsa free

Colasoft Capsa [15] is an open-source network traffic and protocol analyser with a rich set of features [16]. It provides graphical statistics for global network as well as specific nodes in a dashboard tab. A graphical display of both broadcast and multicast packets [17] traversing the network is obtained with Capsa [17]. It also gives the packet count for TCP and UDP traffic along with the amount of TCP FIN and TCP RST sent. It allows for saving the displayed data in CSV format. Protocol statistics include features like sent and received packets and bytes as well average packets per second.

3 Classification of network traffic using machine learning in Weka

This section describes the main classification algorithms used and how the classification was performed using Weka.

3.1 Classification algorithms

Machine Learning techniques help to identify different applications and protocols in a network by grouping them based on packet flow parameters. These include minimum, maximum and mean number of packets, packet length, flow duration, traffic rate, volume, etc. ML classification techniques can be of two types: supervised and unsupervised [5].

In supervised learning technique, a complete labeled data set is required to classify unknown classes. This dataset is used to train the model which will predict output responses in a new set of data. Unsupervised machine learning approach does not constitute complete labeled data. This technique cannot be applied directly for classification because the output is unknown.

A set of 8 ML algorithms is used for this work. They are Naive Bayes, Bayesian Network, Multi-Layer Perceptron, Support Vector Machine, Radial Basis Function Neural network, KNN, bagging and C4.5 Decision Tree. A detailed description of these techniques can be found in references [18–22].

3.2 Classification using Weka Toolkit

The classification process was performed using Weka toolkit [23]. The latter is used as a data mining tool to implement IP traffic classification with ML algorithms. The overall process involves feeding the feature sets containing information about each sample with their labels into the

machine learning algorithm to generate a classifier model. The efficiency and accuracy of the obtained model to capture a pattern is then determined by comparing the labels generated by the model for the inputs in a test set with the correct labels for those inputs. This classification process is illustrated in Fig. 1.

The performance of the classifiers was based on the following criteria:

(i) Classification Accuracy.
Accuracy is the simplest metric deployed to evaluate a classifier. It gives the percentage of inputs in the test set that the classifier correctly labeled.

$$Accuracy = \frac{\sum TP + \sum TN}{\sum Totalno.ofsamples} \quad (1)$$

where True Positives (TP): relevant items correctly identified as relevant. True Negatives (TN): irrelevant items correctly identified as irrelevant.

To define the remaining parameters, False Positives (FP) and False Negatives (FN) are also used. FP denotes irrelevant items incorrectly identified as relevant, while FN represents relevant items incorrectly identified as irrelevant.

(ii) Precision (P).
Precision indicates the number of items identified as relevant and is given by:

$$Precision = \frac{TP}{TP + FP}. \quad (2)$$

(iii) Recall (R).
Recall value indicates the number of relevant items that are identified.

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

(iv) The F-Measure (or F-Score).
This combines the precision and recall to give a single score, also called the harmonic mean of the precision and recall.

$$F-Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (4)$$

(v) The confusion Matrix.

The confusion matrix summarises the performance of a multi-class classifier. If P denotes the first class and N is the second, the confusion matrix can be represented as shown in Fig. 2.

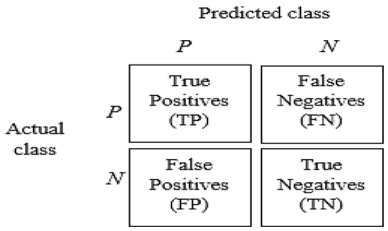

Fig. 2 Confusion matrix [24]

4 Experimental set-up and testing procedures

The overall set-up for the experiments is shown in Fig. 3. The tests were performed on a PC connected to a Wi-Fi network. For this project, a 2.70 GHz Intel core i5 CPU with 4 GB RAM and 64-bit Windows 10 Operating System workstation was used. The network interface discovery feature in PRTG, Capsa and Wireshark was enabled to monitor IP traffic for the Wi-Fi network on the PC.

Data was captured for a duration of 30 min in intervals of 15 s for the on-going session of each application and state. For the classification of applications, three datasets of 700 samples each were built from raw data captured from the three monitoring tools and were saved as CSV files.

Streaming, uploading, downloading and idle state were considered for further classification. The size of datasets for state classification was of 470 samples.

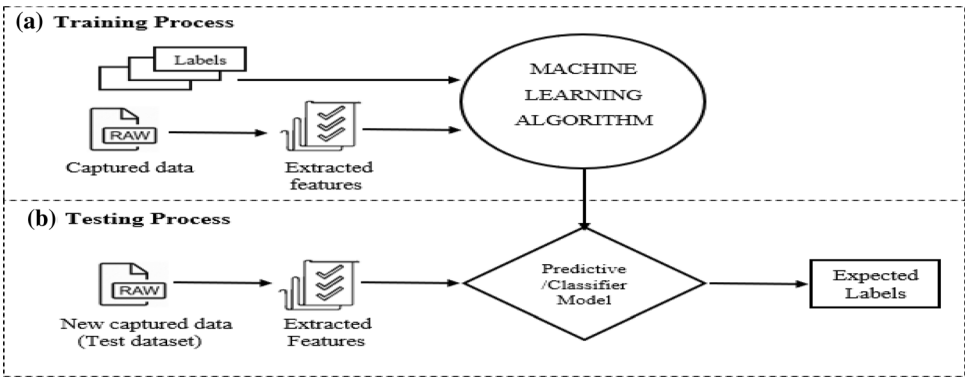

Fig. 1 ML Classification in Weka

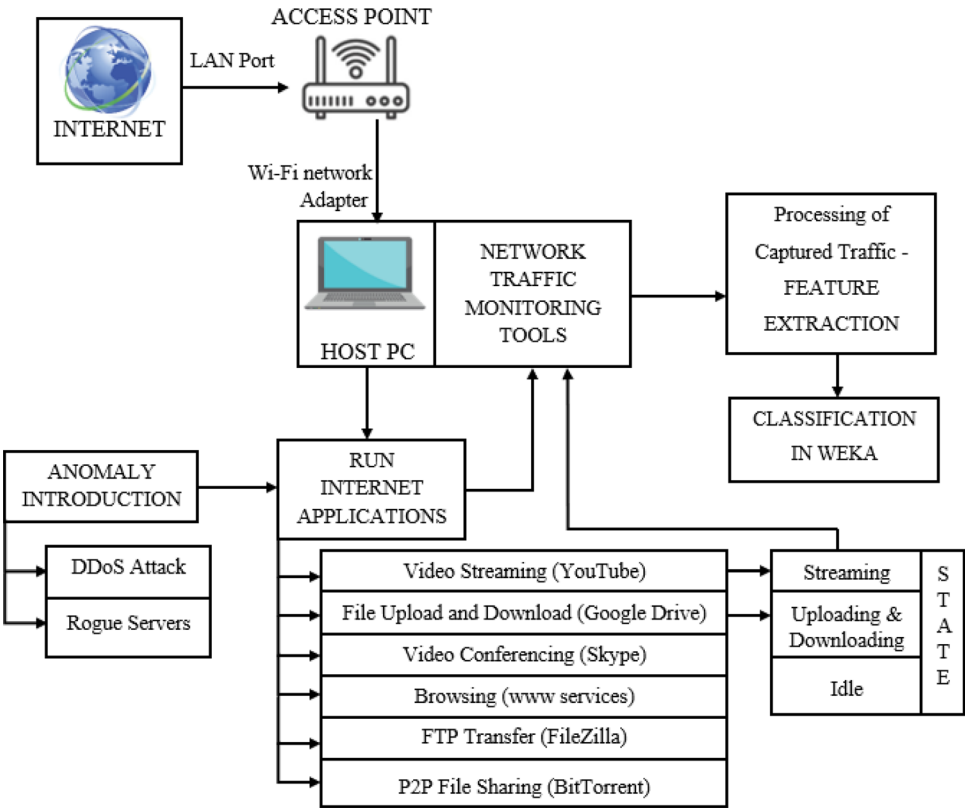

Fig. 3 Overall implemented system

As for anomaly classification, datasets of 700 samples with three classes labeled as normal, DDoS and Rogue Servers were used. The 'normal' class was obtained by running Internet applications under normal conditions.

The network traffic monitoring tools as well as the Weka classifier application were run on the PC. Classification algorithms were used to classify six different internet applications namely video streaming on YouTube, File download and upload via Google Drive, Browsing, Video Conferencing, FTP transfer and P2P File sharing. The experiment was performed for four different states in which the PC can be set namely, streaming, uploading, downloading and idle. Moreover, classification of two different anomalies namely DDoS attack and Rogue Servers were also investigated. Details of these testing conditions are given in the following sub-sections.

4.1 Applications and protocols

Most internet applications operate according to the Client/Server model in the Application layer of the TCP/IP model. A client is a device that requests information and server is the device that responds to the request. Format of requests and responses between clients and servers are generally defined by Application layer protocols [25].

The applications monitored in this study are hereafter described.

(i) Online (Real-Time) Streaming.

Real time streaming implies sending audio or video data and played by the receiver on the other end with a negligible and consistent delay. This process can involve only a sender and a receiver, hence point-to-point, or one sender and several receivers, called broadcast. Real-time streaming prioritises accurate and quick delivery of data. For this purpose, User Datagram Protocol (UDP) is used to deliver continuous information and avoid re-sending dropped packets as does TCP [26].

Application used: YouTube.

(ii) Upload and Download via e-mail.

Upload is referred to as the transfer of data from a client to a server while data transfer from server to client is called download. During e-mail operations, the Mail User Agent (MUA) or e-mail client applications are usually used. The e-mail client uses Post office Protocol (POP) to receive e-mail messages from an e-mail server and the Simple Mail Transfer Protocol (SNMP) allows e-mail to be sent from either a client or a server.

Application used: Google Drive.

(iii) Video Conferencing.

Video conferencing via the Internet makes use of the Voice Over Internet protocol (VoIP). VoIP technology

enables voice to be transmitted over the Internet as a digital signal [27].

Application used: Skype.

(iv) Web Browsing.

World Wide Web (WWW) services are accessible through a web server. To establish a connection to a web service on a server, the web browser uses the Hypertext Transfer Protocol (HTTP). The process involves running background services by the server to allow for requested files by the client to be available. The browser converts the information received by the server into a plain text or HTML format and displays it for the user.

Application used: WWW services.

(v) FTP Transfer.

File Transfer Protocol (FTP) enables file transfer between a client and a server. FTP needs to establish two connections between the client and the server for successful transfer. The first connection, consisting of commands and replies, is made to the server by the client and is established on TCP port 21. The second connection is then made over TCP port 20 for actual file transfer.

Application used: FileZilla Server.

(vi) P2P Applications.

A Peer-to-Peer (P2P) application is one where a device can behave as both the client and the server during the same transfer process. P2P implies requesting information off of other computers and not from a server. Therefore, the client is a server and vice versa. Both client and server can set up a connection and have equal priority.

Application used: BitTorrent.

4.2 Network anomalies

Many works have been done in the area of network anomaly detection. This problem is usually approached using Artificial Intelligence and Machine Learning techniques.

In this project, 2 types of anomalies are investigated: (1) Distributed Denial of Service (DDoS) and (2) Rogue Servers.

The DDoS attack refers to the disruption of normal traffic of a server by bombarding the targeted server with excessive Internet traffic, eventually jamming the network infrastructure and prevent desired traffic from reaching its destination [28]. For this research work, a DDoS attack is generated through a code written in JavaScript which serves to open an infinite number of tabs on Google Chrome continuously, and hence preventing the user to access the network and servers as well as jamming the network infrastructure and slows down or completely shut down the operation of Internet applications. The code was run on the NetBeans IDE.

Rogue servers are set up on a network which serve to disrupt access to a target server. It makes use of the Dynamic Host Configuration Protocol (DHCP), a network protocol that allows an IP address from a given range of numbers to be automatically assigned to a computer by a server. Rogue server attacks are launched by attackers in the form of Sniffing and Reconnaissance attacks, among others [29, 30]. To create rogue servers in the system under study, a code was written in JavaScript which consists of three rogue servers and each made to listen to allocated ports 50,300, 50,302 and 50,305 respectively. These port numbers form part of the dynamic/private port range of 49,152–65,535. The code was run on Node.js.

5 Results and analysis

5.1 Features extracted from monitoring tools

Table 1 shows the list of features obtained from the monitoring three monitoring tools.

The performance and efficiency of the 8 ML classifiers were tested for the classification of applications, states and anomalies. The applications are YouTube, Google Drive, Skype, Browsing, FTP and BitTorrent. The states are four and include Downloading, Uploading, Streaming and Idle state. As for the classification of anomalies, a feature set with 3 classes labeled as ‘normal’, ‘DDoS’ and ‘Rogue servers’ is used.

5.2 Classification of applications

The classification accuracy (A), training time (T) and root mean square error (RMSE) obtained from the ML algorithms for classification of applications characterized by traffic flow features extracted from PRTG, Capsa and Wireshark are tabulated below (Table 2).

For application classification based on PRTG features, the KNN algorithm gives the best accuracy which is 98.7%. However, it has the highest training time at 16.3 s. RBF neural network is considered as the best classifier in this case with a classification accuracy of 98.3%, a training time of 0.99 s and root mean squared error of 7.1%.

For the Capsa feature set, Naïve Bayes best classifies the applications with an accuracy of 100%, shortest training time of 0.03 s and zero error.

KNN classifier has the highest classification accuracy but its high training time of 7.9 s makes it inappropriate also for classification of applications using Wireshark features. Bayes Net gives an accuracy of 99.6% and it has

Table 1 List of extracted features from PRTG, Wireshark and Capsa

Features extracted	PRTG	Wireshark	Capsa
Total number of packets	✔	✔	✔
Total volume (bytes)	✔	✔	✔
Transmission speed (bps)	✔		✔
Average speed (bps)			✔
Broadcast bytes			✔
No. of broadcast packets			✔
Multicast bytes			✔
No. of multicast packets			✔
Packets per second (pps)	✔		✔
Average packets per second (pps)			✔
Bytes received	✔		✔
No. of packets received	✔		✔
Speed of traffic-in (bps)	✔		
Bytes sent	✔		✔
No. of packets sent	✔		✔
Speed of traffic-out (bps)	✔		
Sent/received bytes			✔
Sent/received packets			✔
Packet duration (s)		✔	
% packet loss	✔		
Ping time (ms)	✔		
DNS response time (ms)	✔		
TCP conversation count			✔
Protocol		✔	
Source port		✔	
Destination port		✔	
% System CPU load	✔		
% System health	✔		
% Available memory	✔		
IP Bytes			✔
IP packets no.			✔
IP average bps			✔
IP average pps			✔
TCP bytes			✔
TCP packets no.			✔
TCP average bps			✔
TCP average pps			✔
UDP bytes			✔
UDP packets no.			✔
UDP average bps			✔
UDP average pps			✔

the shortest training time, making it the most efficient application classifier for the case of Wireshark.

The most appropriate feature set for classifying applications is illustrated in Fig. 4 for the comparison of classification accuracy.

KNN has the best accuracy for all three tools but at the cost of high training times. Besides, most ML classifiers best classify applications characterized by features from Capsa, except for SVM classifier which gives a higher accuracy with Wireshark feature.

Figure 5 below displays the precision and recall values obtained from the ML classifiers in the classification of applications with PRTG features.

It can be seen that Google Drive is the best classified application in terms of precision. All 8 ML classifiers give precision value of 1, representing 100% precision. Since Google Drive application was considered for uploading a 150 MB video file onto the server, the traffic generated included larger amount of sent Bytes and packets and smaller volume of incoming data as compared to the other applications, and therefore it could be easily distinguished by the ML classifiers.

It can also be seen that Bayes Net, Naïve Bayes and RBF classifiers give 100% precision for BitTorrent, Browsing, FTP and Google Drive application while RBF network gives 100% precision for all applications except for YouTube.

SVM is the worst classifier with very low recall value for most applications compared to other ML schemes. Bayes Net gives 100% recall for BitTorrent, FTP, Skype and YouTube. MLP gives same for BitTorrent, FTP, Google Drive and Skype. However, RBF network gives a 100% recall for 5 applications and hence chosen as the best ML classifier in this case.

The behavior of RBFNN and SVM can be further explained by their respective confusion matrices as in Table 3.

It can be clearly observed that all applications are correctly identified as themselves with the case of RBFNN while SVM fails to distinguish between the different applications. 25 YouTube samples, 17 Google Drive samples and 28 Skype samples are classified as Browsing. It also classifies 38 FTP and 14 BitTorrent instances as Browsing. This validates the high FP rate of 62.2% obtained with Browsing application as tabulated above.

Figure 6 shows the precision and recall values obtained from the ML classifiers with Capsa features. Google Drive proved to be the best classified application, denoted by 100% precision and recall by all ML classifiers, except SVM. Naïve Bayes, RBFNN, MLP and KNN give 100% recall and precision for all applications considered. SVM on the other hand has a very poor performance. Combining the percentage accuracy from Table 2, Naïve Bayes is the best classifier for the classification of Capsa features-based applications.

The behavior of Naïve Bayes and SVM can be further explained by their respective confusion matrices as in the Table 4.

Table 2 Evaluation metrics for classification of applications

Feature set	PRTG			CAPSA			WIRESHARK		
Parameters	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)
Classifiers									
Bayes Net	97.8	0.03	8.2	99.1	0.11	5.3	99.6	0.02	8
Naive Bayes	96.6	0.09	10.1	100	0.03	0	50	0	34.4
SVM	48.7	0.53	41.3	22.2	0.34	50.9	90.3	0.28	17.9
MLP	97.8	1.71	7.5	100	3.78	0.7	84.8	0.72	20.6
RBFNN	98.3	0.99	7.1	100	0.16	0	77.3	5.5	22.5
KNN	98.7	16.3	6.4	100	26.2	0	100	7.9	0
Bagging	97.8	0.02	8.2	98.7	0.19	5.8	97.4	0.08	10.4
C4.5 DT	97.4	0.09	9.1	98.7	0.06	6.4	99.1	0.38	5.3

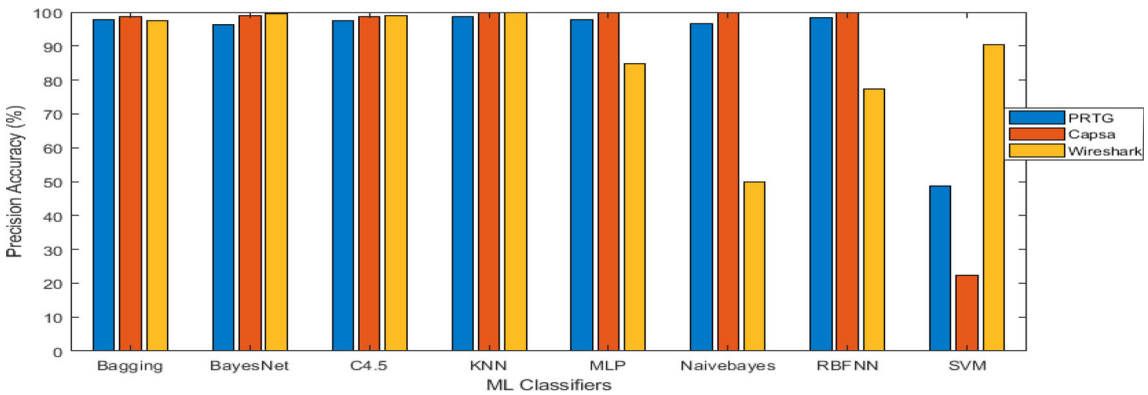

Fig. 4 Comparison of classification accuracy between PRTG, Capsa and Wireshark

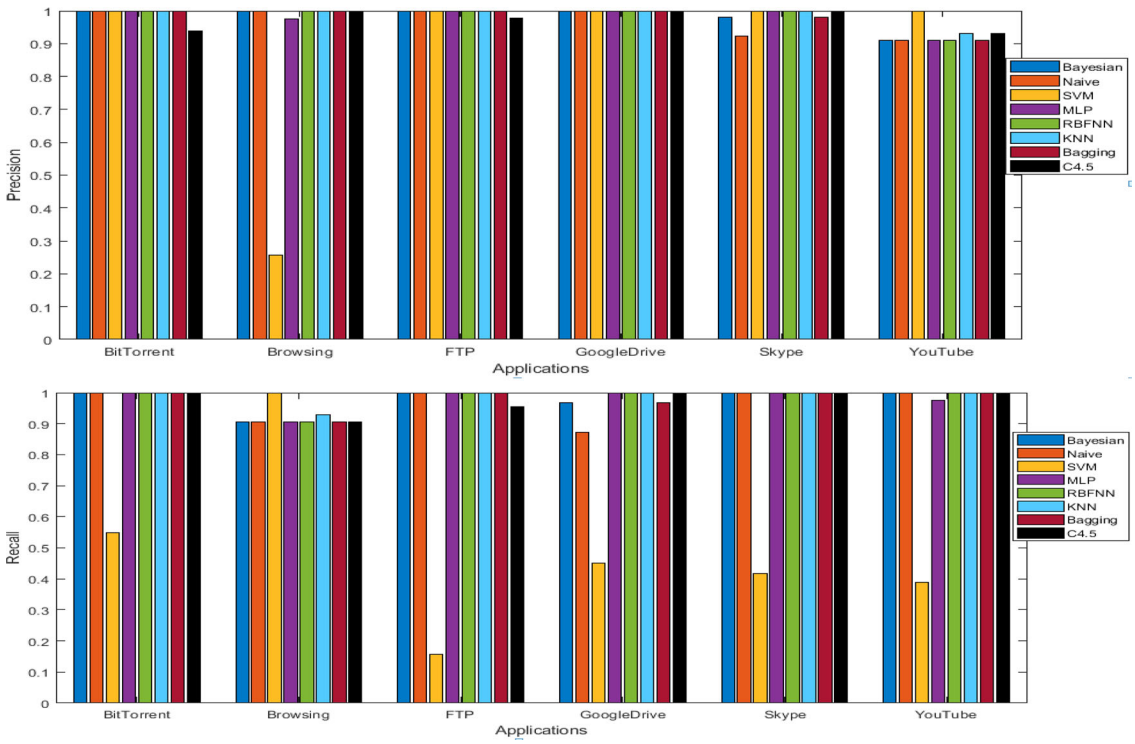

Fig. 5 Precision and Recall of ML classifiers from classification of applications using PRTG dataset

Table 3 Confusion matrices for RBFNN and SVM

Best classifier: RBFNN	Worst classifier: SVM
a b c d e f <-- classified as 41 0 0 0 0 0 | a = YouTube 0 31 0 0 0 0 | b = GoogleDrive 0 0 48 0 0 0 | c = Skype(VoIP) 4 0 0 38 0 0 | d = Browsing(WWW) 0 0 0 0 45 0 | e = FTP 0 0 0 0 0 31 | f = BitTorrent(P2P)	a b c d e f <-- classified as 16 0 0 25 0 0 | a = YouTube 0 14 0 17 0 0 | b = GoogleDrive 0 0 20 28 0 0 | c = Skype(VoIP) 0 0 0 42 0 0 | d = Browsing(WWW) 0 0 0 38 7 0 | e = FTP 0 0 0 14 0 17 | f = BitTorrent(P2P)

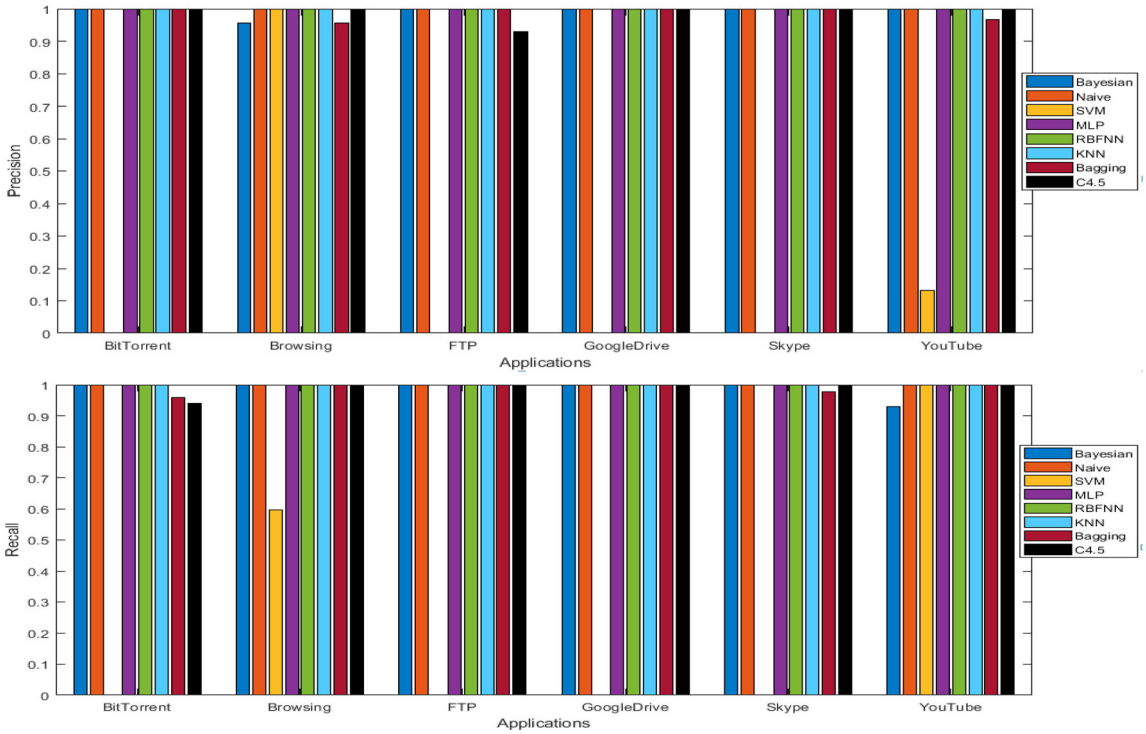

Fig. 6 Precision and Recall of ML classifiers from classification of applications using Capsa dataset

Table 4 Confusion matrices for Naive Bayes and SVM

Best classifier: Naïve Bayes	Worst classifier: SVM
a b c d e f <-- classified as 28 0 0 0 0 0 | a = YouTube 0 33 0 0 0 0 | b = GoogleDrive 0 0 45 0 0 0 | c = Skype(VoIP) 0 0 0 42 0 0 | d = Browsing(WWW) 0 0 0 0 40 0 | e = FTP 0 0 0 0 0 50 | f = BitTorrent(P2P)	a b c d e f <-- classified as 28 0 0 0 0 0 | a = YouTube 33 0 0 0 0 0 | b = GoogleDrive 45 0 0 0 0 0 | c = Skype(VoIP) 17 0 0 25 0 0 | d = Browsing(WWW) 40 0 0 0 0 0 | e = FTP 50 0 0 0 0 0 | f = BitTorrent(P2P)

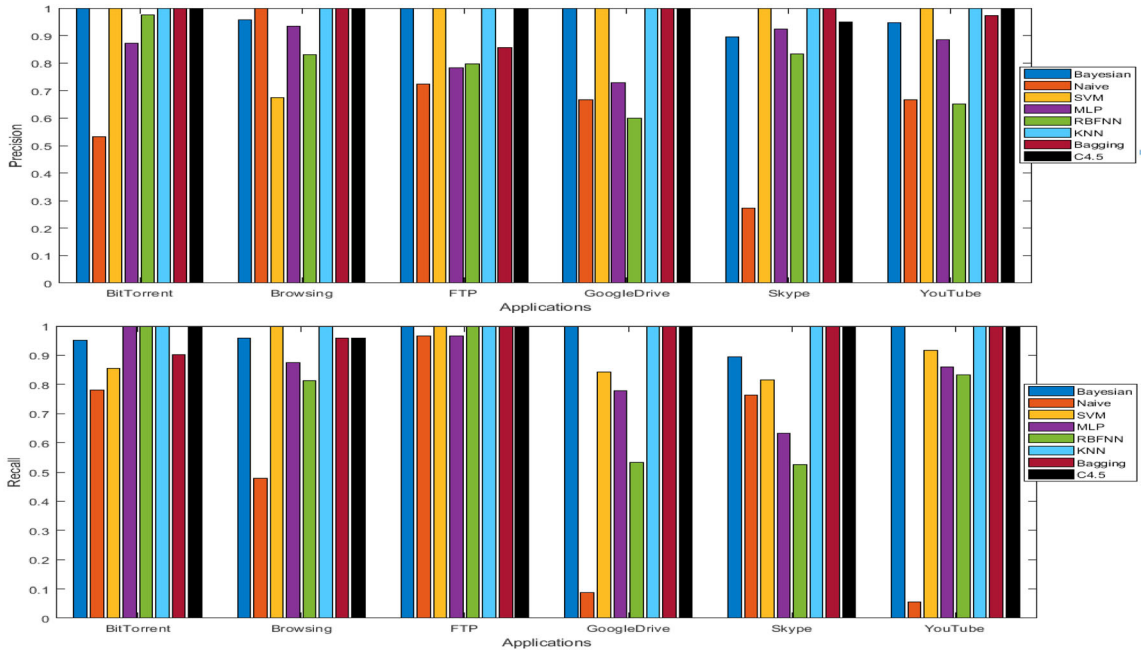

Fig. 7 Precision and Recall of ML classifiers from classification of applications using Wireshark dataset

All 238 instances in the train Capsa set are correctly identified by Naïve Bayes. SVM on the other hand classifies all Google Drive, Skype, FTP and BitTorrent samples as YouTube. This validates the high FP rate of 0.881 (88.1%) obtained with YouTube application.

The precision and recall values obtained from the ML classifiers with Wireshark features are as shown in Fig. 7. BitTorrent application is best classified in terms of Precision while FTP is best classified in terms of Recall. An F-score of 100% is obtained by 5 classifiers for both applications. BitTorrent is easily distinguished by the classifiers since it consists of Peer-to-Peer sharing and involves more UDP packets than other applications.

It can also be observed that C4.5 Decision Tree demonstrates high efficiency by giving 100% precision for five of the six applications. It also gives the best recall value for all applications except for Browsing. On the other hand, Naïve Bayes classifier gives the worst performance in terms of both precision and recall. Although Bayes net gave the highest classification accuracy, it is less reliable than C4.5 in terms of recall and precision.

The classification accuracy and training time of C4.5 was found to be 99.1% and 0.38 s respectively, making it an acceptable ML scheme for the classification of applications.

The behavior of C4.5 DT and Naïve Bayes can be further explained by their respective confusion matrices as shown in Table 5.

Most applications are correctly identified as themselves with the case of C4.5 DT. Only 2 Browsing instances are mistaken to be YouTube application. On the contrary, Naïve Bayes largely fails to distinguish between the different applications.

Table 6 gives the average values of the performance evaluation metrics; i.e., TP and FP rate, precision (P),

Table 5 Confusion matrices for C4.5 DT and Naive Bayes

Best classifier: C4.5 DT							**Worst classifier: Naïve Bayes**						
a	b	c	d	e	f	<-- classified as	a	b	c	d	e	f	<-- classified as
36	0	0	0	0	0	| a = YouTube	2	2	12	0	1	19	| a = YouTube
0	45	0	0	0	0	| b = GoogleDrive	0	4	31	0	8	2	| b = GoogleDrive
0	0	38	0	0	0	| c = Skype(VoIP)	1	0	29	0	2	6	| c = Skype(VoIP)
0	0	2	46	0	0	| d = Browsing(WWW)	0	0	24	23	0	1	| d = Browsing(WWW)
0	0	0	0	30	0	| e = FTP	0	0	1	0	29	0	| e = FTP
0	0	0	0	0	41	| f = BitTorrent(P2P)	0	0	9	0	0	32	| f = BitTorrent(P2P)

Table 6 Average precision and recall values for overall classification of applications

Classifiers	Feature set														
	PRTG					CAPSA					WIRESHARK				
	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F
Bayes Net	0.98	0.005	0.98	0.98	0.97	0.99	0.002	0.99	0.99	0.99	0.96	0.007	0.96	0.96	0.96
NB	0.97	0.008	0.97	0.96	0.96	1.00	0	1.00	1.00	1.00	0.50	0.09	0.65	0.50	0.45
SVM	0.48	0.11	0.87	0.48	0.51	0.22	0.10	–	0.22	–	0.90	0.02	0.93	0.90	0.90
MLP	0.98	0.004	0.98	0.97	0.97	1.00	0	1.00	1.00	1.00	0.84	0.03	0.85	0.84	0.84
RBFNN	0.98	0.003	0.99	0.98	0.98	1.00	0	1.00	1.00	1.00	0.77	0.05	0.77	0.77	0.76
KNN	0.99	0.003	0.99	0.98	0.98	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00
Bagging	0.98	0.005	0.98	0.97	0.97	0.98	0.002	0.98	0.98	0.98	0.97	0.004	0.97	0.97	0.97
C4.5 DT	0.98	0.005	0.97	0.97	0.97	0.98	0.003	0.98	0.98	0.98	0.99	0.002	0.99	0.99	0.99

recall (R) and F-measure (F) for the overall classification of the six applications.

It further confirms that Capsa feature set is the best for classification of applications. KNN, MLP, Naïve Bayes and RBF Network give 100% precision and recall for Capsa while none of them gives ideal values for PRTG. As for Wireshark, recall and precision value of 1 are only obtained with C4.5 and KNN. Thus, it can be deduced that for application classification, better classification performance is portraited by ML algorithms when a dataset with more features is used. Also, solely the Capsa dataset contains detailed information about IP, TCP and UDP traffic, which largely contribute to proper classification of Internet applications.

5.3 Classification of states

The classification accuracy, training time and root mean square error obtained from the ML algorithms in the classification of the four states, namely Downloading, Uploading, Streaming and Idle are tabulated in Table 7.

Figure 8 compares the accuracy of the ML algorithms.

For state classification based on PRTG features, the Bayes net, Naïve Bayes, MLP, KNN and Bagging algorithms give 100% classification accuracy. However, Bayes Net and Naïve Bayes are considered as the best classifiers in this case with a training time of 0.02 s and zero RMSE.

For the Capsa feature set, Naïve Bayes best classifies the states with an accuracy of 100%, shortest training time of 0.02 s and zero error.

The best classifier based on classification accuracy and training time using Wireshark feature set is Bayes Net.

Bayes Net displays the best overall classification performance for all three tools. Besides, most ML classifiers best classify states characterized by features from PRTG, except for SVM classifier which gives a higher accuracy with Wireshark features.

The precision and recall values obtained from the ML classifiers in the classification of states with PRTG features are summarised in Fig. 9. In terms of precision, it can be seen that Uploading and Downloading are the best classified states. All 8 ML classifiers give precision value of 1, representing 100% precision. It can be difficult to differentiate between uploading and downloading sessions of the same video file as both mainly involve TCP traffic.

Table 7 Algorithms' performance evaluation metrics for state classification

Feature set	PRTG			CAPSA			WIRESHARK		
Parameters	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)
Classifiers									
Bayes Net	100	0.02	0	98.7	0.01	7.9	99.3	0.01	7.5
Naive Bayes	100	0.02	0	100	0.02	0	55.6	0	42.5
SVM	60.2	0.21	44.3	26.2	0.18	60.7	90.6	0.07	21.6
MLP	100	1.02	0.6	100	1.8	0.4	95	0.35	17.8
RBFNN	100	0.1	0	100	0.05	0	64.3	0.29	34.7
KNN	100	13.32	0	100	17.2	0	97.5	10.2	11.1
Bagging	100	0.1	1.2	100	0.07	4.2	95.6	0.02	13.5
C4.5 DT	98.7	0.06	7.9	98.7	0.03	7.9	97.5	0.02	10.9

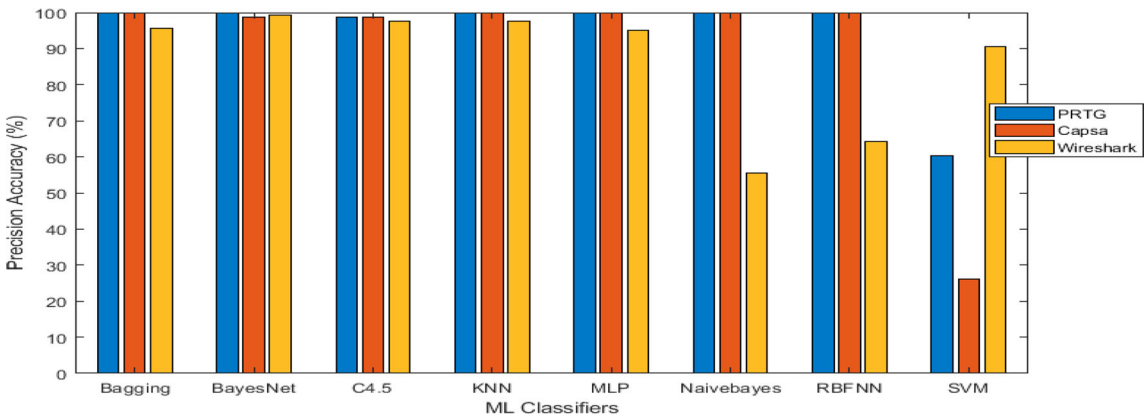

Fig. 8 Comparison of classification accuracy between PRTG, Capsa and Wireshark for state classification

However, downloading implies larger volume of incoming traffic and less outgoing traffic and vice versa for uploading state. It can be therefore deduced that PRTG provides concise features that fully contribute to distinguish and classify these two states.

As for the other two states, i.e., Streaming and Idle, they are perfectly classified by all algorithms, except C4.5 and SVM for Idle and Streaming respectively.

From the recall chart, it is clearly seen that SVM is the worst classifier with very low recall value for 3 out of 4 states. Bayes Net, Naïve Bayes, MLP, RBFNN, KNN and Bagging are equally good classifiers and they give 100% recall and precision for all four states. Streaming is the best classified state in terms of recall.

The behavior of Bayes Net and SVM can be further explained by their respective confusion matrices as shown in Table 8.

All 160 instances in the train PRTG set are correctly identified by Bayes Net. SVM on the other hand classifies almost half of Downloading samples, 20 Idle state samples and 24 out of 44 Uploading samples as Streaming. However, no streaming samples are classified as other states. That is why a bad FP rate of 51.6% but a good TP rate of

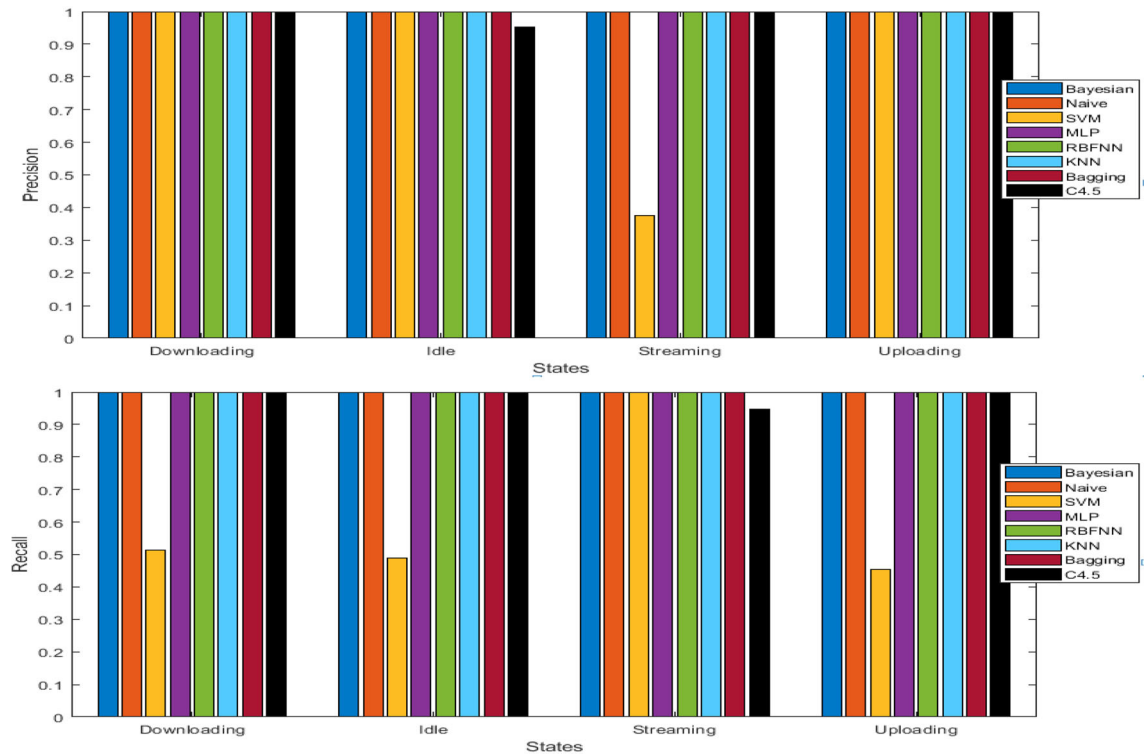

Fig. 9 Precision and Recall of ML classifiers from classification of states using PRTG dataset

Table 8 Confusion matrices for Bayes Net and SVM

Best classifier: Bayes Net	Worst classifier: SVM
a b c d <-- classified as 37 0 0 0 | a = Downloading 0 44 0 0 | b = Uploading 0 0 38 0 | c = Streaming 0 0 0 41 | d = Idle	a b c d <-- classified as 19 0 18 0 | a = Downloading 0 20 24 0 | b = Uploading 0 0 38 0 | c = Streaming 0 0 21 20 | d = Idle

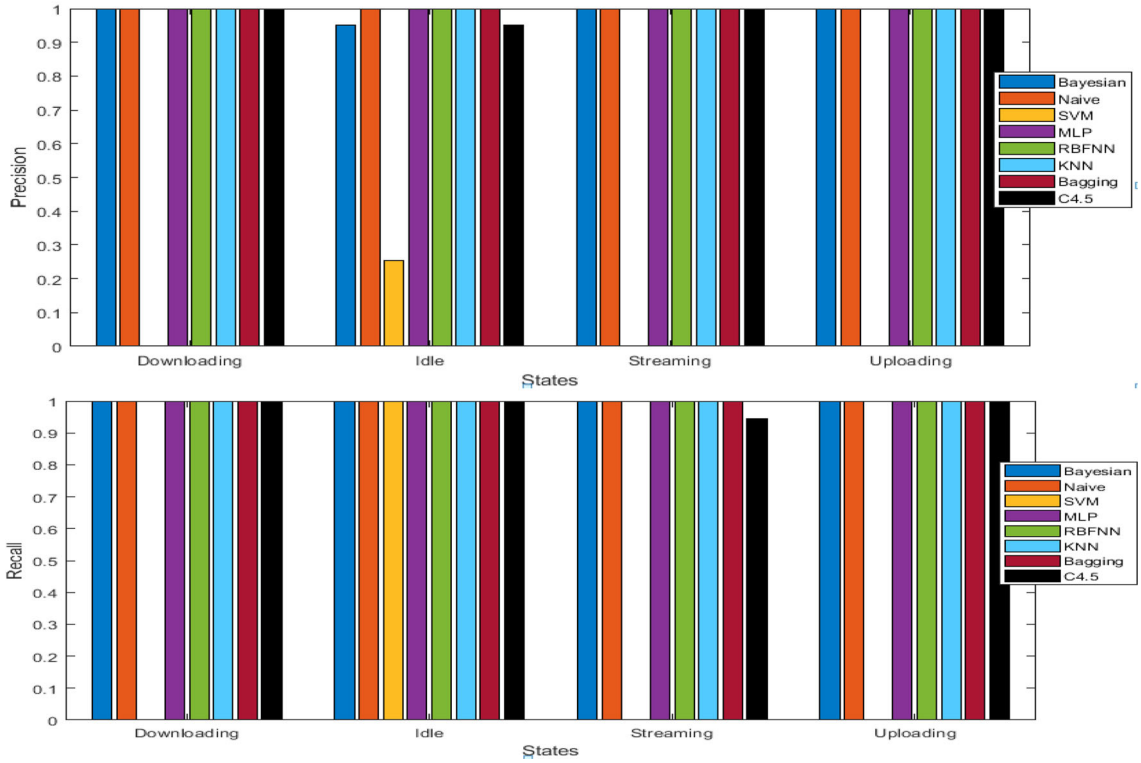

Fig. 10 Precision and Recall of ML classifiers from classification of states using Capsa dataset

100% are obtained for the SVM classifier with PRTG feature set.

The precision and recall values obtained from the ML classifiers with Capsa features are summarized in Fig. 10. SVM gives precision and recall only for the Idle state. In addition to that, a 100% recall is achieved with all ML algorithms for the Idle state Thus, features extracted from Capsa are best suited for the classification of Idle state. The least amount of traffic is generated when the PC is idle and is not being used for Internet applications. It is therefore easier to differentiate Idle state from the other 3 states.

Naïve Bayes, RBF network, MLP, KNN and Bagging give 100% precision and recall for all four states.

The behavior of the best classifiers and SVM can be further explained by their respective confusion matrices as shown in Table 9.

SVM is unable to classify all states. 158 samples out of the 160 samples present in the test set are identified as Idle state. This is denoted by the high False Positive rate of 98.3% given by SVM for the Idle state in the table above.

The precision and recall values respectively obtained from the ML classifiers in the classification of states with Wireshark features are shown in Fig. 11. Bayes Net gives 100% precision for downloading, uploading and idle states and 100% recall for downloading, streaming and uploading states, making it the most efficient algorithm for state classification using Wireshark features. On the other hand, lowest precision and recall are obtained with Naïve Bayes and RBF networks, hence explaining their low classification accuracy values.

During Idle state, protocols traversing the network are mainly ARP and ICMP requests as compared to the other

Table 9 Confusion matrices

Best classifier: NB, RBF, MLP, KNN, Bagging	Worst classifier: SVM
a b c d <-- classified as 44 0 0 0 | a = Downloading 0 39 0 0 | b = Uploading 0 0 37 0 | c = Streaming 0 0 0 40 | d = Idle	a b c d <-- classified as 0 0 0 44 | a = Downloading 0 0 0 39 | b = Uploading 0 0 2 35 | c = Streaming 0 0 0 40 | d = Idle

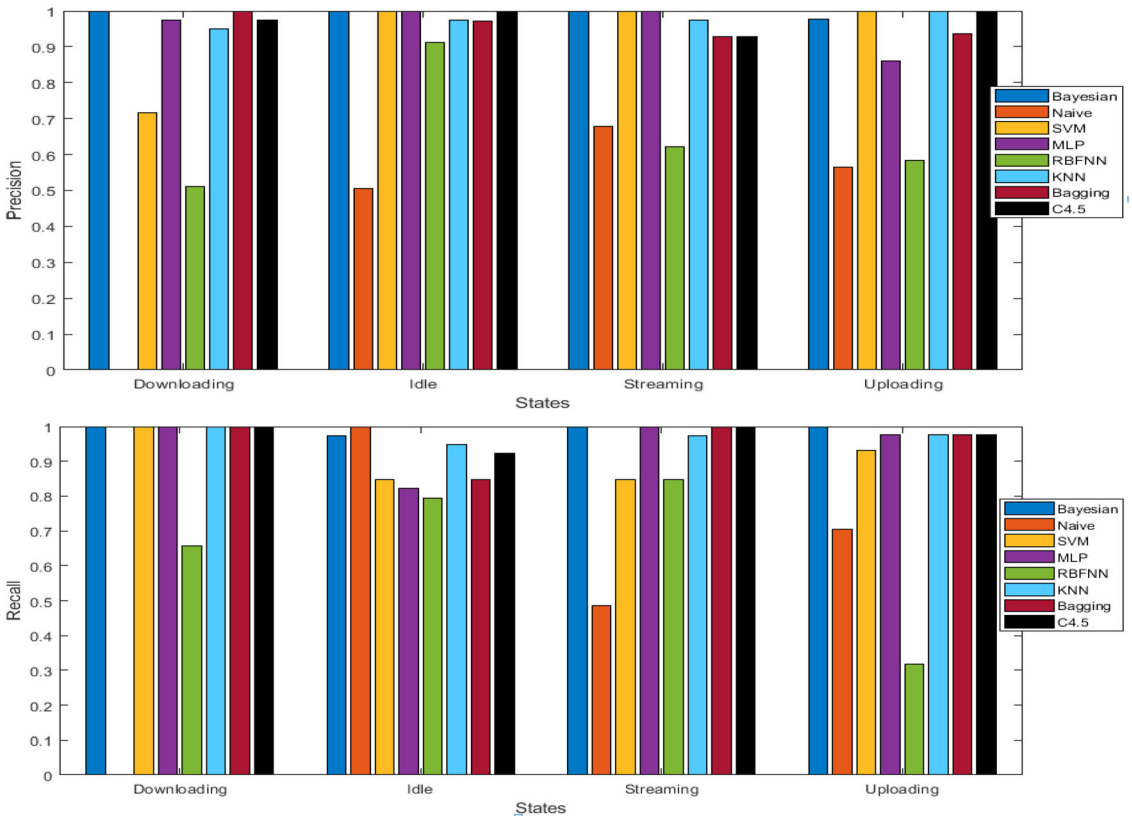

Fig. 11 Precision and Recall of ML classifiers from classification of states using Wireshark dataset

states which involve TCP and UDP packets. Since Wireshark characterizes samples by protocols, the Idle state is the easiest identified and classified one in terms of Precision and Recall by the ML algorithms.

Table 10 gives the Confusion matrices for the Bayes Net and Naive Bayes classifiers.

It can be clearly observed that all states are correctly identified as themselves with the case of Bayes Net while SVM fails to distinguish between the different states. Downloading samples are classified as Uploading, Idle and Streaming. 13 Uploading samples are classified as Streaming and Idle. 20 Streaming samples are classified as Uploading and Idle. This validates the high FP rate of 31.4% obtained with Idle state, 20.7% obtained with Uploading and 7.4% with Streaming.

Table 11 gives the average values of the performance evaluation metrics for the overall classification of the 4 states.

It can be concluded that state classification using Wireshark gives the poorest performance among the three monitoring tools. Moreover, 100% precision and recall are obtained with 6 out of 8 ML classifiers using PRTG compared to 5 classifiers when using Capsa. It can be confirmed that the PRTG feature set is the best for state classification. The PRTG set contains 17 features while that of Capsa consists of 30 features. However, three important features are provided by PRTG that enhance the performance of ML classifiers in the classification of states. They are system health, CPU load and available memory.

Table 10 Confusion matrices for Bayes Net and Naive Bayes

Best classifier: Bayes Net	Worst classifier: Naïve Bayes
a b c d <-- classified as 38 0 0 0 | a = Downloading 0 44 0 0 | b = Uploading 0 0 39 0 | c = Streaming 0 1 0 38 | d = Idle	a b c d <-- classified as 0 20 5 13 | a = Downloading 0 31 4 9 | b = Uploading 0 4 19 16 | c = Streaming 0 0 0 39 | d = Idle

Table 11 Average precision and recall values for overall classification of states

Classifiers	Feature set														
	PRTG					CAPSA					WIRESHARK				
	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F
Bayes Net	1.00	0	1.00	1.00	1.00	0.99	0.004	0.99	0.99	0.99	0.99	0.002	0.99	0.99	0.99
NB	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00	0.56	0.15	–	0.56	–
SVM	0.61	0.12	0.85	0.61	0.63	0.26	0.25	–	0.26	–	0.91	0.03	0.93	0.91	0.91
MLP	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00	0.95	0.02	0.96	0.95	0.94
RBFNN	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00	0.64	0.12	0.66	0.64	0.63
KNN	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00	0.98	0.008	0.98	0.98	0.98
Bagging	1.00	0	1.00	1.00	1.00	1.00	0	1.00	1.00	1.00	0.96	0.02	0.96	0.96	0.96
C4.5 DT	0.99	0.004	0.99	0.99	0.99	0.99	0.004	0.99	0.99	0.99	0.98	0.008	0.98	0.98	0.98

5.4 Classification of anomalies

The classification accuracy, training time and root mean square error obtained from the ML algorithms in the classification of two types of anomalies, namely DDoS and Rogue Servers, along with a class of normal traffic, characterized by traffic flow features extracted from PRTG, Capsa and Wireshark are tabulated in Table 12.

For anomaly classification based on PRTG features, the MLP and KNN algorithms give the best accuracy which is 99.1%. However, KNN has the highest training time at 46 s. MLP is therefore considered as the best classifier due to its lower training time of 2.88 s and RMSE of 7.7%.

For the Capsa feature set, MLP best classifies the anomalies with an accuracy of 99.1%, but at the cost of a relatively high training time of 4.1 s. The second best classifier is C4.5 Decision Tree. It gives a classification accuracy of 96.6%, considerably shorter training time of 0.33 s and RMSE of 14.4%.

The Bagging classifier has the highest classification accuracy for anomaly classification using Wireshark features. It gives an accuracy of 75.2% and training time of 0.27 s. Moreover, Naïve Bayes is the only classifier that takes 0 s to train the model using Wireshark dataset but its low classification accuracy of 52.5% makes it inefficient for classification.

Table 12 Classification accuracy of ML algorithms for classifying anomalies

Feature set	PRTG			CAPSA			WIRESHARK		
Parameters	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)	A (%)	T (s)	RMSE (%)
Classifiers									
Bayes Net	98.7	0.26	9.7	94.5	0.01	17.6	56.7	0.01	40.4
Naive Bayes	60.9	0.08	47.6	81.0	0.01	32.8	52.5	0	46.2
SVM	57.5	0.72	53.1	43.6	0.38	61.2	73.9	0.17	41.6
MLP	99.1	2.88	7.7	99.1	4.1	6.6	58.4	0.59	38.1
RBFNN	81.1	0.59	31.4	82.7	0.36	27.7	42.0	0.06	45.6
KNN	99.1	46	7.4	100	57.6	0	71.4	16.7	43.6
Bagging	97.1	0.25	13.1	95.3	0.13	12.4	75.2	0.27	33.9
C4.5 DT	97.4	0.03	12.9	96.6	0.33	14.4	72.6	0.08	37.2

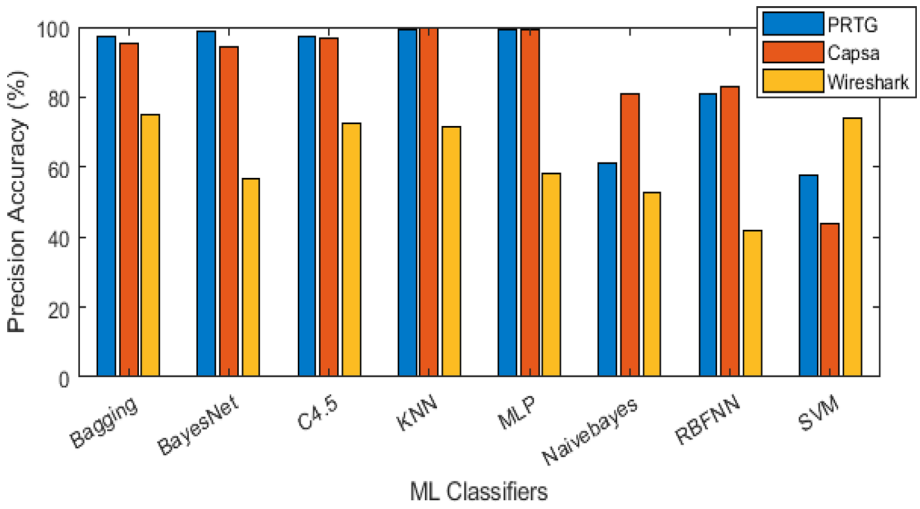

Fig. 12 Comparison of classification accuracy between PRTG, Capsa and Wireshark

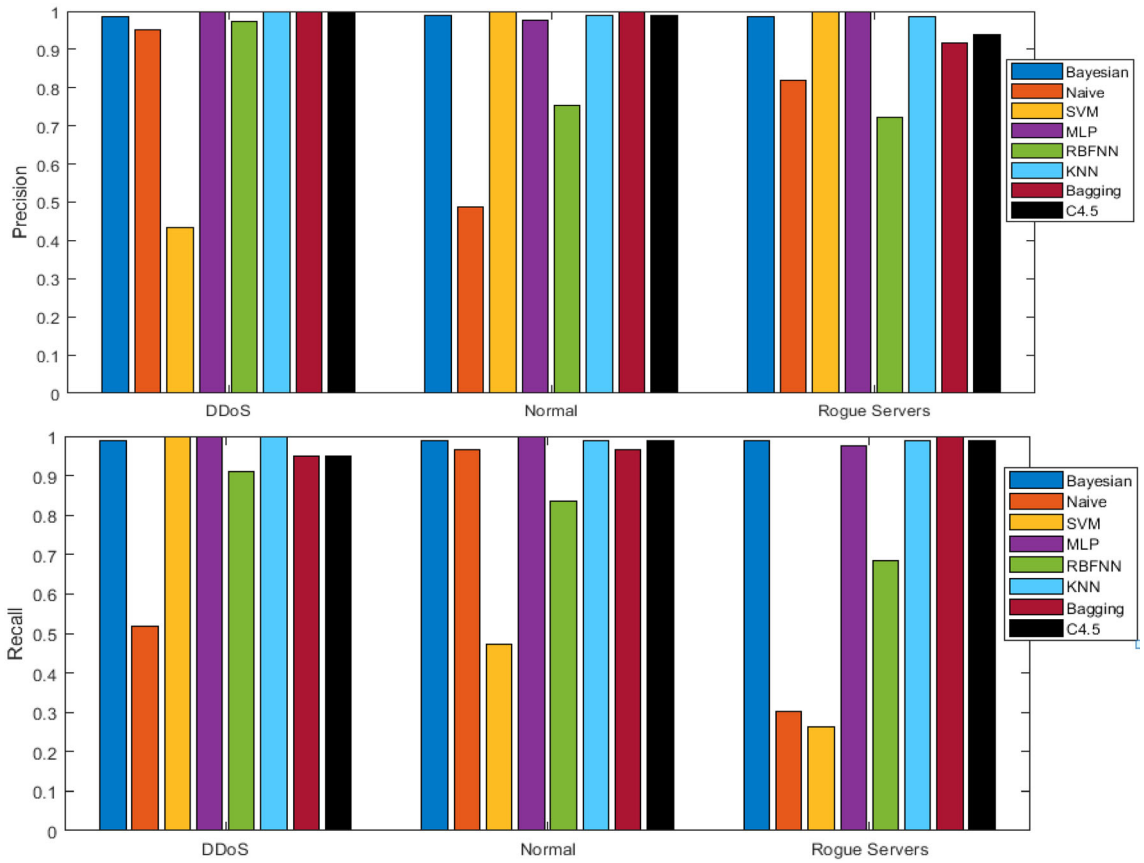

Fig. 13 Precision and Recall of ML classifiers from classification of anomalies using PRTG dataset

The most appropriate feature set for classifying anomalies is illustrated in Fig. 12 comparison of classification accuracy.

The precision and recall values obtained from the ML classifiers in the classification of anomalies with PRTG features are summarised in Fig. 13.

An overall view of the two above bar charts shows that MLP is the classifier with 100% precision for DDoS and Rogue servers, and 100% recall for DDoS and Normal class samples. On the other hand, lowest precision and recall are obtained with Naïve Bayes. DDoS samples are relatively better classified in terms of recall and precision with PRTG features. During DDoS attack, a significantly

Table 13 Confusion matrices for MLP and Naive Bayes

Best classifier: MLP					Worst classifier: Naïve Bayes				
a	b	c	<-- classified as		a	b	c	<-- classified as	
85	0	0	|	a = Normal	82	0	3	|	a = Normal
0	77	0	|	b = DDoS	35	40	2	|	b = DDoS
2	0	74	|	c = RogueServers	51	2	23	|	c = RogueServers

larger volume of traffic was recorded, and hence, makes it easily distinguishable from the other classes.

The behavior of MLP and Naïve Bayes can be further explained by their respective confusion matrices as in Table 13.

It can be clearly observed that MLP classifies all instances without fail except for 2 rogue server samples that it wrongly identifies as Normal samples. On the other side, many false positives are obtained with SVM. It mistakes 35 DDoS samples and 51 Rogue samples for Normal. This is why an FP rate of 56.2% is obtained with Naïve Bayes for Normal class.

The precision and recall values obtained from the ML classifiers in the classification of anomalies with Capsa features are summarised in Fig. 14.

From Fig. 14, it can be seen that the Normal class is the best classified in terms of precision. 100% precision is obtained with SVM, MLP, KNN and C4.5. The second diagram reveals that Rogue server anomaly is better classified in terms of recall.

Overall, KNN algorithm exhibits best classification result and SVM displays the poorest classification performance with very low recall value for 2 out of 3 classes (Table 14).

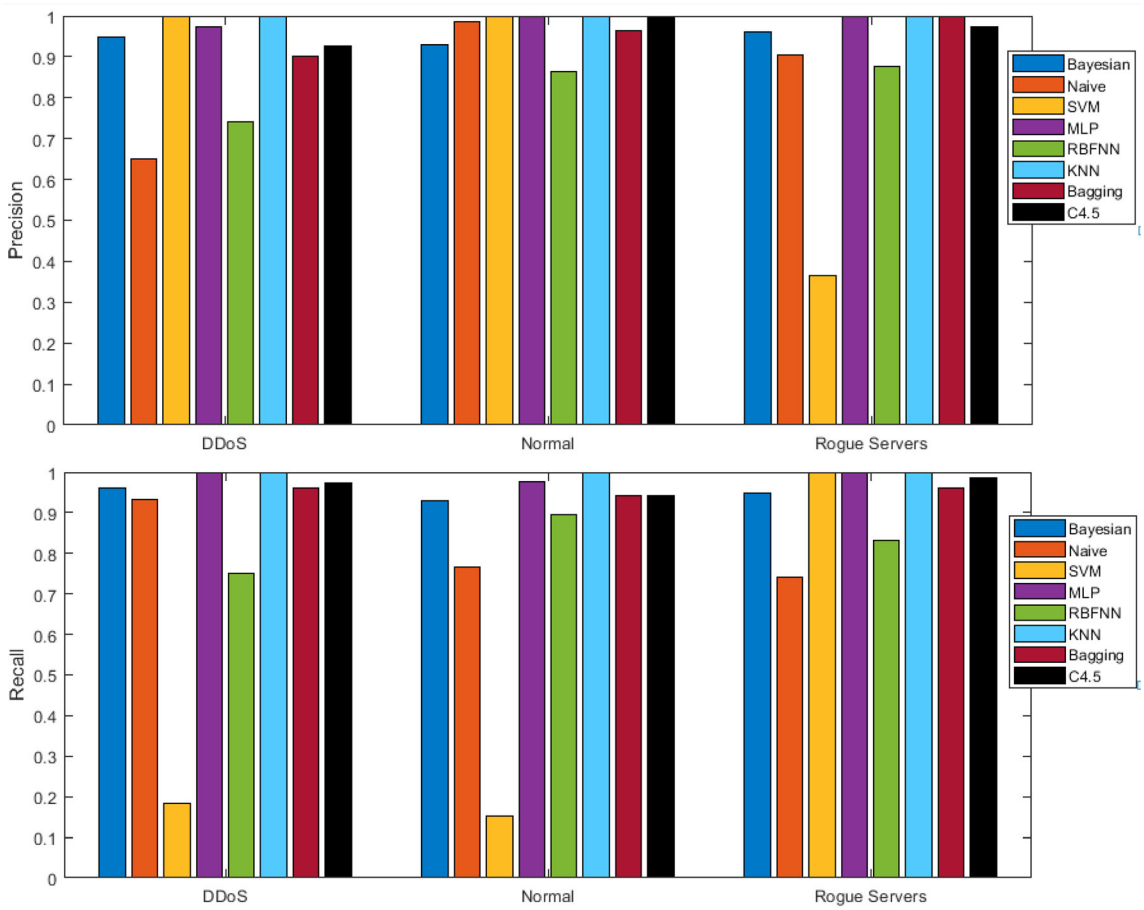

Fig. 14 Precision and Recall of ML classifiers from classification of anomalies using Capsa dataset

Table 14 Confusion matrices for KNN and SVM

Best classifier: KNN	Worst classifier: SVM
a b c <-- classified as 85 0 0 | a = Normal 0 76 0 | b = DDoS 0 0 77 | c = RogueServers	a b c <-- classified as 13 0 72 | a = Normal 0 14 62 | b = DDoS 0 0 77 | c = RogueServers

KNN exhibits ideal classification results contrarily to SVM which wrongly classifies Normal and DDoS instances as Rogue Servers, which explains its high FP rate of 83.2%.

The precision and recall values obtained from the ML classifiers in the classification of anomalies with Wireshark features are summarised Fig. 15.

Using Wireshark feature set, DDoS is found to be better classified in terms of precision and Rogue Servers in terms of recall. An important factor to consider here is that Wireshark provides the Source and Destination ports features. Since Rogue Server attack implies connection onto other designated ports, Rogue Server samples have greater chances of being recognized from other samples, and therefore best classified by the ML algorithms. The classification performance of the ML algorithms varies from one another and no best algorithm can be deduced from the above two figures. However, it can be clearly observed without further analysis that the Wireshark feature set is not the best option for classification of anomalies.

Table 15 gives the average values of the performance evaluation metrics for the overall classification of the 3 anomaly classes.

It can be concluded that anomaly classification using Wireshark gives the poorest performance among the three monitoring tools.

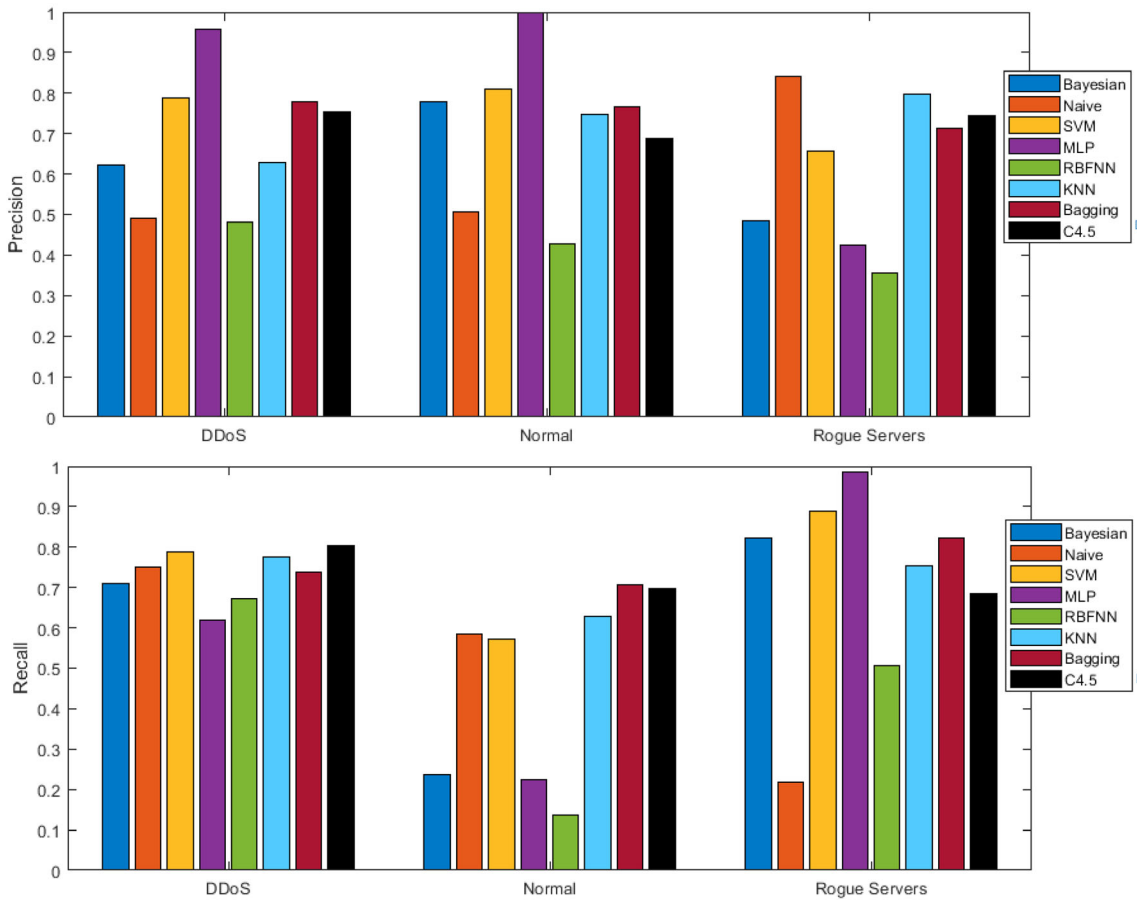

Fig. 15 Precision and Recall of ML classifiers from classification of anomalies using Wireshark dataset

Table 15 Average precision and recall values for overall classification of anomalies

Classifiers	Feature set														
	PRTG					CAPSA					WIRESHARK				
	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F	TP rate	FP rate	P	R	F
Bayes Net	0.99	0.006	0.99	0.99	0.99	0.95	0.03	0.95	0.95	0.95	0.57	0.20	0.64	0.57	0.53
NB	0.61	0.22	0.75	0.61	0.59	0.81	0.09	0.85	0.81	0.82	0.53	0.25	0.60	0.53	0.49
SVM	0.58	0.20	0.82	0.58	0.56	0.44	0.27	0.80	0.44	0.37	0.74	0.13	0.76	0.74	0.74
MLP	0.99	0.005	0.99	0.99	0.99	0.99	0.004	0.99	0.99	0.99	0.58	0.18	0.81	0.58	0.56
RBFNN	0.81	0.09	0.82	0.81	0.81	0.83	0.09	0.83	0.83	0.83	0.42	0.27	0.42	0.42	0.38
KNN	0.99	0.004	0.99	0.99	0.99	1.00	0.00	1.00	1.00	1.00	0.71	0.14	0.72	0.71	0.72
Bagging	0.97	0.01	0.97	0.97	0.97	0.95	0.02	0.96	0.95	0.95	0.75	0.12	0.76	0.75	0.75
C4.5 DT	0.98	0.01	0.98	0.98	0.98	0.97	0.02	0.97	0.97	0.97	0.73	0.14	0.73	0.73	0.73

Moreover, both PRTG and Capsa features result in approximately same performance of ML classifiers. Only MLP and KNN gives 100% precision and recall. Therefore, both feature extraction tools generate features that are suited for the classification of anomalies.

6 Conclusion

The aim of this paper was to capture Internet traffic from web applications using three traffic monitoring tools (PRTG, Colasoft Capsa and Wireshark) and to deploy eight Machine Learning algorithms for classification of six applications and four states derived from them. The states included Downloading, Uploading, Streaming and Idle states. Two anomalies, namely DDoS and Rogue Server attacks were also generated during traffic capture and were classified using Weka Toolkit. it was noted that Capsa allowed for extraction of the largest number of features, followed by PRTG. The classification results obtained showed that the performance of the ML classifiers varies in each case. It was further observed that Capsa feature set was best suited for classification of applications due to its large number of features. PRTG feature set outperformed that of Capsa in the classification of States. An important implication on this observation is that the contribution of the individual features in classification is more relevant than the overall number of features actually present in a dataset. Finally, ML algorithms gave the poorest performance in the classification of anomalies. A possible explanation would be the presence of only 3 classes and the high level of similarities between them. On an overall perspective, classification based on Wireshark feature set displayed the worst results. Additionally, the SVM classifier gave the poorest performance in the overall classification of Internet traffic. This validates the fact that SVM is largely affected by irrelevant and noisy samples. On the other hand, KNN showcased the highest classification accuracy in most cases but it takes significantly high time to train the classifier model. The Naïve Bayes algorithm can be chosen as an alternative for its robustness to irrelevant samples. This study makes conspicuous that feature selection is an imperative step in the classification of IP traffic. The main limitation encountered in this work is that due to resource constraints, network traffic capture was carried out for short intervals of time, resulting in less samples for classification. The above observations finally pave the way to conclude that ML classification is a reliable technique for analysis of Internet traffic, given the appropriate set of features. Interesting future works will be to optimize the performance of ML algorithms by using larger number of samples and to perform a deeper analysis on the Capsa feature set by deducing the generating cost of individual features and eliminating those which barely contribute to classification performance, and hence reducing network resource consumption.

Acknowledgements The authors would like to thank the University of Mauritius for providing the necessary facilities to conduct this research.

References

1. Joshi P, Bhandari A, Jamunkar K, Warghade K, Lokhande P (2016) Network traffic analysis measurement and classification using Hadoop. Int J Adv Res Comput Commun Eng. https://doi.org/10.17148/IJARCCE.2016.5360
2. Mellia M (2010) Traffic monitoring and analysis: second international workshop, TMA, 2010, Zurich, Switzerland, April 7, 2010. In: Proceedings, computer communication networks and telecommunications volume 6003 of Lecture Notes in Computer Science, Springer, Berlin. ISSN 0302-9743
3. Srinivasa KG, Siddesh GM, Srinidhi H (2018) Network data analytics: a hands-on approach for application development. In: Computer communications and networks series, 1st edn. Berlinl: Springer. 2018 edition 27 Apr 2018

4. Parsaei MR, Sobouti MJ, Khayami SR, Javidan R (2017) Network traffic classification using machine learning techniques over software defined networks. IJACSA 8(7):220–225
5. Shafiq M, Xiangzhan Y, Asif AL, Lu Y, Nabin KK, Foudil A (2016) Network traffic classification techniques and comparative analysis using machine learning algorithms. In: 2nd IEEE international conference on computer and communications (ICCC), Chengdu China, 14–17 Oct 2016
6. Singh K, Agrawal S (2011) Comparative analysis of five machine learning algorithms for IP traffic classification. In: International conference on emerging trends in networks and computer communications (ETNCC), 22–24 Apr 2011, Udaipur, India
7. Agrawal S, Jaspreet K, Sohi BS, Machine learning classifier for internet traffic from academic perspective. In: International conference on recent advances and future trends in information technology (iRAFIT2012), Proceedings published in International Journal of Computer Applications® (IJCA)
8. Zhoua D, Yana Z, Fua Y, Yaoa Z (2018) A survey on network data collection. J Netw Comput Appl 116:9–23. https://doi.org/10.1016/j.jnca.2018.05.004
9. Iglesias F, Zseby T (2015) Analysis of network traffic features for anomaly detection. Mach Learn 101(1–3):59–84. https://doi.org/10.1007/s10994-014-5473-9
10. PRTG Network Monitor, Paessler, [Online]. Available: https://www.paessler.com/prtg
11. PRTG Manual: Key Features, Paessler, 2019. [Online]. Available: https://www.paessler.com/manuals/prtg/key_features. Accessed 18 Nov 2018
12. Lammle T (2016) CCNA Routing and switching complete study guide: Exam 100-105, Exam 200-105, Exam 200-125, Sybex; 2 edition. 17 Oct 2016
13. Wireshark (online) Available: https://www.wireshark.org/
14. Chappell L (2017) Wireshark® 101: Essential Skills for Network Analysis, Second Edition: Wireshark Solution Series [Print Replica] Kindle Edition, 2017, Amazon Digital Services LLC
15. Capsa Standard 11, Colasoft (2018) (online). Available: https://www.colasoft.com/landing/capsa_std.php
16. Monitor Network Traffic, Colasoft (2018) (online). Available: https://www.colasoft.com/capsa/monitor-network-traffic.php. Accessed 2 Mar 2019
17. Zheng J, Jamalipour A (2009) Broadcasting, multicasting, and geocasting. Wiley, New York, pp 145–172. https://doi.org/10.1002/9780470443521.ch5
18. Cheng J, Greiner R (2001) Learning bayesian belief network classifiers: algorithms and system. In: Stroulia E, Matwin S (eds) Advances in artificial intelligence. Canadian AI 2001. Lecture notes in computer science (lecture notes in artificial intelligence), vol 2056. Springer, Berlin
19. Huang D, Guan G, Zhou J, Wang H (2018) Network-based naive Bayes model for social network. Sci China Math 61(4):627–640. https://doi.org/10.1007/s11425-017-9209-6
20. Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29(2–3):131–163. https://doi.org/10.1023/A:1007465528199
21. Kruse R, Borgelt C, Braune C, Mostaghim S, Steinbrecher M (2016) Multilayer perceptrons. In: Computational intelligence. Texts in computer science. Springer, London. https://doi.org/10.1007/978-1-4471-7296-3_5
22. Zhongqi W, Bo Y, Yonggang K, Yuan Y (2016) Development of a prediction model based on RBF neural network for sheet metal fixture locating layout design and optimization. Comput Intell Neurosci. https://doi.org/10.1155/2016/7620438
23. Weka 3: Data Mining Software in Java, The University of Waikato, [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/
24. Susmaga R (2004) Confusion matrix visualization. In: Kłopotek MA, Wierzchoń ST, Trojanowski K (eds) Intelligent information processing and web mining Advances in Soft Computing, vol 25. Springer, Berlin. https://doi.org/10.1007/978-3-540-39985-8_12
25. Rao UH, Nayak U (2014) Understanding networks and network security. In: The InfoSec Handbook. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-6383-8_9
26. Rodriguez-Gil L, Orduña P, García-Zubia J, López-de-Ipiña D (2018) Interactive live-streaming technologies and approaches for web-based applications. Multimed Tools Appl 77:6471. https://doi.org/10.1007/s11042-017-4556-6
27. Ransome JF, Rittinghouse JW (2005) Voice over Internet Protocol (VoIP) Security. Digital Press. https://doi.org/10.1016/B978-1-55558-332-3.X5000-6
28. Mahjabin T, Xiao Y, Sun G, Jiang W (2017) A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int J Distrib Sens Netw. https://doi.org/10.1177/1550147717741463
29. Razaque A, Elleithy K (2013) Controlling attacks of rogue dynamic host configuration protocol (DHCP) to improve pedagogical activities in mobile collaborative learning (MCL) environment. J Commun Comput Eng 3(1):15–29
30. Univeristy of Waikato, Attribute-Relation File Format (ARFF), 1 November 2008. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/arff.html. Accessed 4 Nov 2018