packets [33]. Accordingly, burstiness is a measure of the variability of packet arrival spacing for a traffic flow over time. Figure 1 shows how the group of packets forms a burst based on inter-packet arrival times, based on a set threshold (Burst_threshold) that defines the maximum inter-arrival time for two

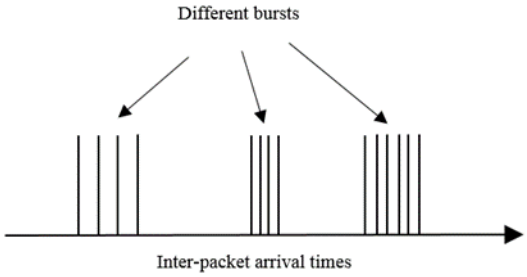

Figure 1. The burstiness concept

packets to belong to the same burst. Initially, some features can be calculated from this figure such as number of bursts per flow, size of bursts and number of packets in bursts. A study was carried out as part of this study to determine the value of Burst_threshold as shown in figure 2. The figure shows the inter-packet arrival times for six applications (i.e., BBC news, Facebook, Google search, Skype, Yahoo mail, YouTube) in msec. Most distributions of the inter-packet arrival times fall under 1 second. Accordingly, the Burst_threshold was set to 1 second. The concept of burst, which is proposed by this work, was implemented in tcptrace tool. The pseudocode in algorithm 1 summarises the estimation of bursts; this code was written in C script as the tcptrace source code. The process of bursts calculation could be illustrated as follow. Firstly, the inter-arrival time between successive packets is computed, if it is less than burst_threshold, a new burst is formed, and some values would be accumulated such as current burst and current session. Otherwise, if the inter-packet arrival time is greater than burst_threshold, it means that the previous burst was finished and new one would be formed and so on. This process was carried out for each direction of the flow and by calculating the number of bursts that were formed per direction.

Algorithm 1: Estimation of packet bursts time

```
Burst threshold= 1s
initialise burst and idle time parameters
while packets arriving
do
 calculate interarrival_time
 if interarrival_time < burst_threshold
  current_burst ++
  current_session ++
 else
  burst_counter ++
  current_burst = 1
 fi
done
```

Also, the parameters for each direction were computed such as number of packets in the total bursts for this direction, size of these packets and duration of the burst. The possible features that could be extracted as output from the pseudocode can be classified into two groups. The first group is related to the burst features that are calculated between packets arrival time within flows. In addition, this group contains two types of burst features which were calculated for either all packets or for only data packets in flow. To distinguish between two types of the first group, we denoted the burst features for only data packets as "data", Table 2 displays the proposed features. The first group features were calculated by writing a script inside tcptrace tool. The second group is related to the burst features that were calculated between flows

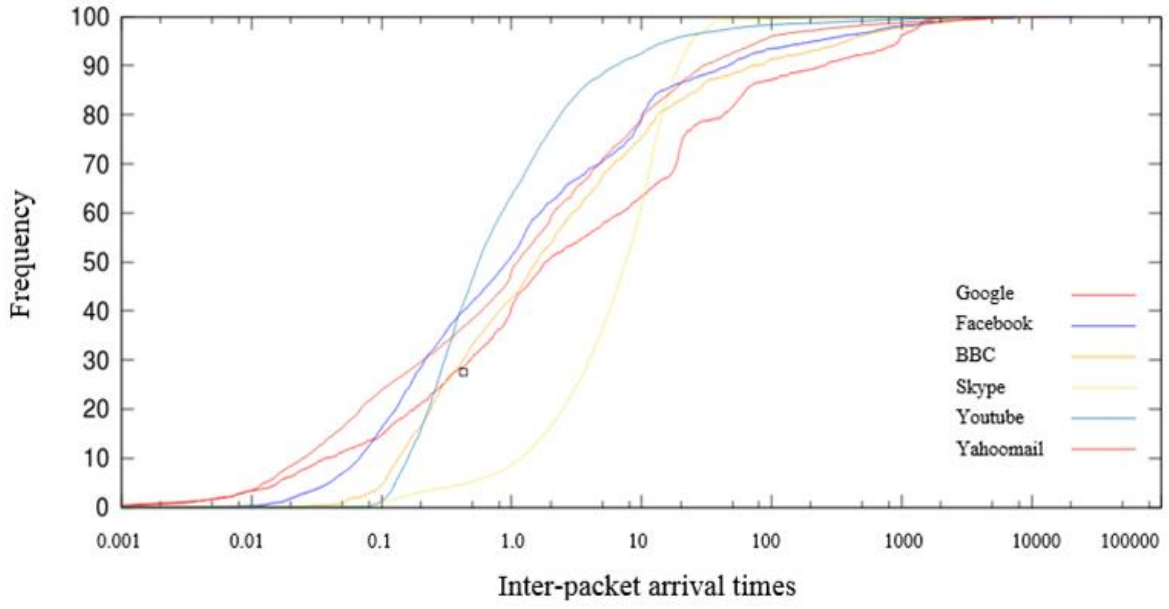

Figure 2. The distribution of inter-packet arrival times (msec)

arrival time. These features are number of bursts between flows and number of flows in bursts. The second group is calculated through writing a script in R. Both of these groups were fed to the C5.0 classifier as will explain in the next section.

Table 2. Proposed parameters for each direction

Features	Description
Burst_no_a, Burst_no_b Burst_data_no_a, Burst_data_no_b	Number of bursts that contain all packets or only data packets
Pkt_count_a, Pkt_count_b Pkt_data_count_a, Pkt_data_count_b	Number of packets in bursts for all packets or only data packets
Burst_size_bytes_a, Burst_size_bytes_b	Size of bursts in bytes
Burst_duration_a, Burst_duration_b Burst_dur_data_a, Burst_dur_data_b	The duration of bursts for all packets or only data packets
Inter_arrival_time_burst_a Inter_arrival_time_burst_b	The time duration of bursts divided by the total packets

The architecture of the proposed system is illustrated in figure 3, showing the components of application scheme. Primarily, the data was captured from six users that accessed six Internet applications, which are frequently used by the users. The raw traffic was analysed using tcptrace tool that has been modified to generate the proposed features. Tcptrace is a tool that manipulates with packets as input and generates flows as output. The tool was exploited by writing the script inside it that computed the burstiness features. Afterword, R scripts were written for pre-processing the output of the previous stage and create additional features derived from the proposed ones, as shown in the figure. Finally, five statistical operation (i.e., maximum, minimum, mean, median, and standard deviation) were applied only upon the basic and packets burst features. The aim of these processes is to summarize the output of all features in one row for each operation. Hence, it reduced the database of sessions that led to decrease in time consuming. The combined set of newly generated features was fed into a C5.0 ML algorithm to derive a traffic classifier. The data were divided into two parts, the first part was for training and to build the classifier model while the second part was for testing. More details are explained in the following sub sections.

4. Evaluation

The proposed method was evaluated by utilizing C5.0 decision trees algorithm. The classifier was built based on data that were captured from six users. Each user was asked to browse six applications (i.e., BBC news, Facebook, Google searching, Skype, Yahoo mail and YouTube,). The reason for selecting these applications as they are considered to be the most well-known applications [34].

Table 3. Summary of the data collection

Application	Flows	Duration (h)
BBC news	32,596	15.6
Facebook	5,620	12.9
Google searching	27,640	8.5
Skype	2,632	9.88
Yahoo mail	48,116	10.22
YouTube	11,233	11.3

Table 3 summarizes the data collection of the conducted experiment. The users accessed separately each application for (30) times and each time was for (2-5) minutes. The users were limited to using only a single application at any session and the dump files were accordingly labelled with the name of the accessed application. The large and separated dataset made the training data more robust which enabling the classifier to learn properly. The collected data were split in the following approximate proportion: 65% of samples were used for training the C5.0 classifier, while testing was done using the remaining 35% samples per application.

4.1. Recorded accuracy

The accuracy was calculated for the basic features (set1) that was proposed by previous studies and for the proposed features (set2). Set 1 included the following features: number of data packets, number of flag packets, size of the first packets, time duration, inter-arrival-time, received packets to transmitted packets, received of data packets to transmitted data packets, received of flags packets to transmitted of flags, transmitted of flags packets to transmitted packets and received of flags packets to received packets. While set 2 included the features that were shown in table 2. Moreover, the accuracy was calculated for set 3 which is the combining of two sets.

Table 4. Accuracy of the classifier for feature sets

Feature set	No boost	Boost 10	Boost 100
Set 1	93 %	97.33%	97.5%
Set 2	94.33%	96.83%	96.83%
Set 3	90.7%	97.96%	97.96%

The accuracy for these sets are shown in Table 4 and range between 90-97.96%, the accuracy for basic features exceeded the accuracy obtained by the proposed features. However, the accuracy reached to the highest level when the both sets were combined together. In other words, the proposed parameters enhanced the classifier ability to discriminate the

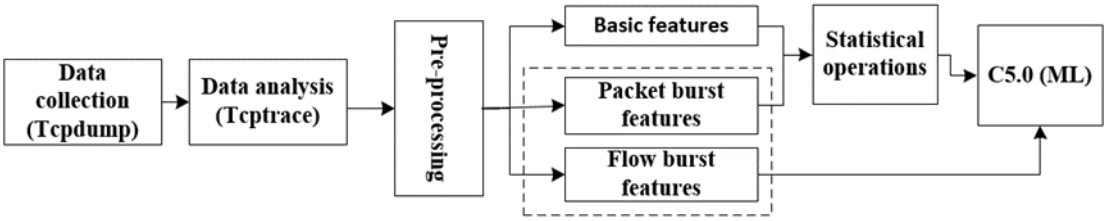

Figure 3. Proposed traffic classification methodology

different traffic that were generated from the applications. The attributes usage (percentage) by the optimal C5.0 in computing the decision tree using feature set 3 is given in Table 5. The table shows the comparison between basic and proposed features. The attributes in interval (100%) reported maximum usage in segregating among the six applications. Also, the parameters in interval (75-99) % showed strongly usage by the classifier. The proposed attributes between packet streams and flows were majority part compared with the basic features which were offered differentiation among applications activities. This is another indicator that the classifier strongly relied on the proposed features because they provide high discrimination between applications.

4.2. Confusion matrix

The accuracy, as presented in the previous section, represents only the ratio of correctly classified instances versus all instances. Therefore, the accuracy does not give us any indication of which class has the error. For further investigating, the performance of the classifier for each class can be represented using confusion matrix. The confusion matrix is a table that displays the visualisation of the classifier performance. It has two dimensions, one for predication instances and the other for the actual instances. Correct classification occurs when the predication instances match with the actual instances. For example, confusion matrix table might be 2*2 or 3*3 as shown in Figure 4, or more than three depends on the number of classes. The row shows the instances in the predicated class while column shows the instances in the actual class. The diagonal of the matrix represents the number of samples that are correctly classified as interest class and called True Positive (TP). The rest of the values in the row of each application are misclassified as the class of interest and called False Positives (FP), and the rest of the values in the column of each application are misclassified as not the class of interest and called False Negatives (FN). The most important point in this measure that it specifies the classifier ability to distinguish one class versus all classes. The performance of the classification model (classifier) on the test data are shown in Table 6 using the confusion matrix. The overall performance of the classifier is considerably high for all applications except Google browsing. Out of a total of sixty samples containing several flows, packet streams and corresponding features, there were four instances classified as (FN) with Yahoo Mail.

Table 5. Attributes Usage in C 5.0 Classifier

Basic features usage (the statistical operations calculated for all flows)	
100%	Mean & median for number of transmitted packets, mean for number of transmitted data packets, mean_flow_duration_b, Max_flow_duration_b, median for the first packets_a & the first packets_b, standard deviation of inter arrival time_b
75-99%	Mean no. of data Packets_b, median for no. of flags packets_a / no. of packets_a, standard deviation for the first Packets_b, Max for inter arrival time_a, mean for the first packet, mean for inter arrival time_b, standard deviation for ratio of no. of packets in both directions, mean for no. of flags packets_b, standard deviation for number of data packets_a, standard deviation for number of packets_a
Proposed features usage (the statistical operations calculated for all flows)	
100%	Mean for number of data burst_a, mean for the inter arrival time_data_b, Max for number of packets in burst_b , Max for data burst_b, Max for data burst duration_b, Max for average of the size of the data burst_b, median for the duration burst_a, median for the inter arrival time_data_a, standard deviation for burst_duration_b, No. of connections for each session, No. of connections in bursts, mean for the ratio of size of burst in both direction, Max for the ratio of the size data burst in both direction

75-99%	Max size of burst_b, Max no. of burst_b, median for the ratio of the burst size in both directions, standard deviation of the no. of packets in burst, mean for the inter arrival time in the burst, Max no. of the data burst_a , standard deviation for the average of the size burst_b, median for the ratio of the data burst size in both directions, Max for the number of packets in burst_a, median for average size of data burst_a, standard deviation for the size burst_a, standard deviation for the size burst_a, standard deviation for the inter arrival time in burst_a, No. of bursts in connections, mean for the size of data burst_b, Max for the inter arrival time in data burst_a, standard deviation for the ratio size burst in both directions, standard deviation for the size of data burst_b, standard deviation for the no. of data packets in burst_a

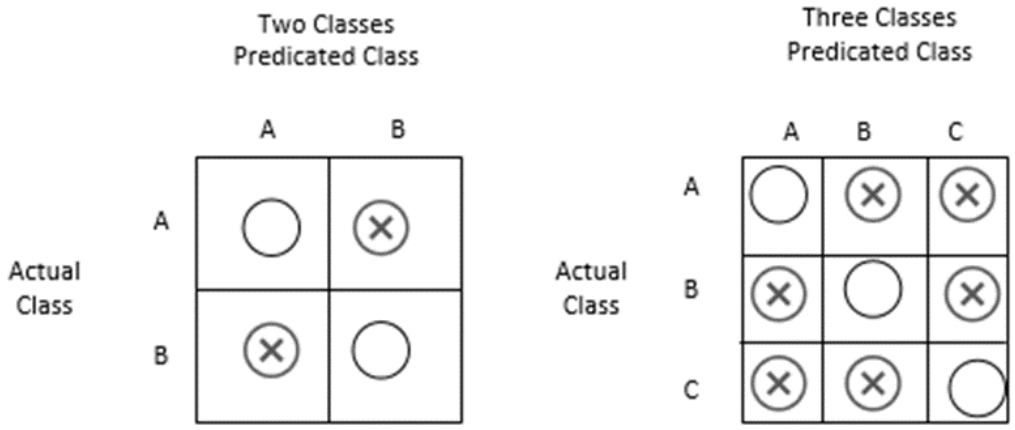

Figure 4. Confusion matrix tables

Table 6. Confusion Matrix table (Feature Set 3)

Application name	**BBC news**	**Facebook**	**Google**	**Skype**	**Yahoo Mail**	**YouTube**
BBC news	**59**	0	0	0	0	0
Facebook	0	**59**	0	0	1	2
Google	0	0	**55**	0	0	0
Skype	0	0	0	**60**	0	0
Yahoo Mail	1	0	4	0	**59**	0
YouTube	**0**	1	1	0	0	58

4.3. Sensitivity and specificity factor

These parameters are a measure of ability of a classifier to identify and discriminate samples of given classes. Sensitivity refers to the derived model's capability to predict the samples that belong to a class or application, while specificity refers to the generated prediction model's capability to mark and differentiate that these samples as not belonging to a given class. Sensitivity therefore avoids the false negative (FN), while specificity avoids the false positives (FP). Accordingly, the sensitivity and specificity can be defined as in the following equations:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

The optimal balance between these factors relies on the type of application that being used. The relationship between these parameters is a trade-off,

when one parameter increase the other decrease. For instance, when the testing occurs in the airport security, the alarm is set on a low-risk items (low specificity) to be likely identify dangerous items and avoid missing any objects (high sensitivity). Both sensitivity and specificity factors for the built classifier are shown in Figure 5 using feature set 3 with a boost factor of 100. The sensitivity of Google browsing was the lowest (97%) due to misclassification with Yahoo Mail. The overall sensitivity ranged above 97%. Also, specificity factor across all six applications was considerably high ranging between (98-100) percent, depicting high segregation ability of the prediction model.

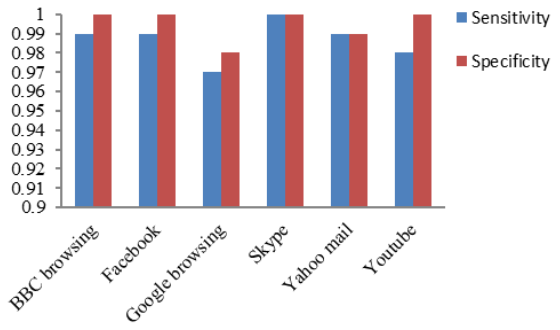

Figure 5. Sensitivity and specificity factors

5. Conclusion

This work proposed an algorithm for applications identification based on a novel feature set. The study exploited the inter-arrival times between packets and flows to generate these features, most specifically burstiness. The selected parameters were evaluated based on data that were collected from six users browsing six applications. Afterwards, the data were analysed using tcptrace tool and fed to the C5.0 classifier for training and testing. The results showed very high accuracy for the proposed method up to 98%. The proposed features set enhanced the ability of the classifier to predict the application type when it added to the previous studies features.

As part of future work, the set of Internet applications considered by the present study will be expanded to include other applications (i.e., online shopping, email, news websites, photo sharing websites, search engines, etc.). Moreover, more experimental work would investigate the visibility of utilizing the inactive time within packets and flows. Different machine learning algorithms would be evaluated to investigate their effects on system performance.

6. References

[1] Cisco, “Cisco visual networking index: forecast and methodology,” 2016–2020 white paper; 2016. Available from: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360 .html.

[2] T. Bakhshi and B. Ghita, “User traffic profiling,” *2015 Internet Technol. Appl. ITA 2015 - Proc. 6th Int. Conf.*, no. November, pp. 91–97, 2015.

[3] M. S. Joe Touch; Eliot Lear, Allison Mankin, Markku Kojo, Kumiko Ono and and A. Z. Lars Eggert, Alexey Melnikov, Wes Eddy, “IANA.” [Online]. Available: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml. [Accessed: 04-Mar-2016].

[4] J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, “BlindBox: Deep Packet Inspection over Encrypted Traffic,” *Proc. 2015 ACM Conf. Spec. Interes. Gr. Data Commun. - SIGCOMM ’15*, pp. 213–226, 2015.

[5] T. Bakhshi and B. Ghita, “On Internet Traffic Classification: A Two-Phased Machine Learning Approach,” *J. Comput. Networks Commun.*, vol. 2016, no. May, 2016.

[6] A. Vlăduţu, D. Comăneci, and C. Dobre, “Internet traffic classification based on flows’ statistical properties with machine learning,” *Int. J. Netw. Manag.*, vol. 27, no. 3, p. e1929, May 2017.

[7] T. Antonio and A. S. Paramita, “Feature selection technique impact for internet traffic classification using Naïve Bayesian,” *J. Teknol.*, vol. 72, no. 5, pp. 141–145, 2015.

[8] J. Cai, Z. Zhang, and X. Song, “An analysis of UDP traffic classification,” *Int. Conf. Commun. Technol. Proceedings, ICCT*, pp. 116–119, 2010.

[9] J. M. Reddy and C. Hota, “Heuristic-Based Real-Time P2P Traffic Identification,” *2015 Int. Conf. Emerg. Inf. Technol. Eng. Solut.*, pp. 38–43, 2015.

[10] S. A. Baset and H. G. Schulzrinne, “An analysis of the Skype peer-to-peer internet telephony protocol,” *Proc. - IEEE INFOCOM*, 2006.

[11] V. C. Español, “Network traffic classification: from theory to practice,” Barcelona University, 2014.

[12] B. Park, Y. Won, J. Chung, M. Kim, and J. W.-K. Hong, “Fine-grained traffic classification based on functional separation,” *Int. J. Netw. Manag.*, vol. 23, no. 5, pp. 350–381, 2013.

[13] S. Sen, O. Spatscheck, and D. Wang, “Accurate, scalable in-network identification of p2p traffic using application signatures,” *Proc. 13th Int. Conf. World Wide Web*, p. 521, 2004.

[14] J. Yan, Z. Wu, H. Luo, and S. Zhang, “P2P traffic identification based on host and flow behaviour characteristics,” *Cybern. Inf. Technol.*, vol. 13, no. 3, pp. 64–76, 2013.

[15] A. Bashir, C. Huang, B. Nandy, and N. Seddigh, “Classifying P2P activity in Netflow records: A case study on BitTorrent,” *IEEE Int. Conf. Commun.*, pp. 3018–3023, 2013.

[16] J. Hurley, E. Garcia-Palacios, and S. Sezer, “Host-Based P2P Flow Identification and Use in Real-Time,” *ACM Trans. Web*, vol. 5, no. 2, pp. 1–27, 2011.

[17] A. Ulliac and B. V Ghita, “Non-Intrusive Identification of Peer-to-Peer Traffic,” in *2010 Third International Conference on Communication Theory, Reliability, and Quality of Service*, pp. 175–183.

[18] A. Hajjar, J. Khalife, and J. Díaz-Verdejo, “Network traffic application identification based on message

size analysis," *J. Netw. Comput. Appl.*, vol. 58, pp. 130–143, 2015.
[19] G. Y. Lazarou, J. Baca, V. S. Frost, and J. B. Evans, "Describing Network Traffic Using the Index of Variability," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1672–1683, 2009.
[20] M. Roughan and S. Sen, "Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification," *Proc. 4th …*, pp. 135–148, 2004.
[21] T. Bujlow, T. Riaz, and J. M. Pedersen, "A method for classification of network traffic based on C5.0 machine learning algorithm," *2012 Int. Conf. Comput. Netw. Commun. ICNC'12*, pp. 237–241, 2012.
[22] P. Pinky and S. E. V. Ewards, "A Survey on IP Traffic Classification Using Machine Learning," *Int. J. Eng. Res. Appl.*, vol. 3, no. 1, pp. 2099–2104, 2013.
[23] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, 2013.
[24] F. Hajikarami, M. Berenjkoub, and M. H. Manshaei, "A Modular Two-layer System for Accurate and Fast Traffic Classification," *Inf. Secur. Cryptol. (ISCISC), 2014 11th Int. ISC Conf. (pp. 149-154). IEEE*, pp. 149–154, 2014.
[25] T. Bakhshi and B. Ghita, "Traffic profiling: Evaluating stability in multi-device user environments," *Proc. - IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2016*, pp. 731–736, 2016.
[26] H. Alizadeh, "Traffic Classification and Verification using Unsupervised Learning of Gaussian Mixture Models," *Meas. Netw. (M&N), 2015 IEEE Int. Work. (pp. 1-6). IEEE*, 2015.
[27] Trianggoro Wiradinata and P. Adi Suryaputra, "Clustering and Principal Feature Selection Impact for Internet Traffic Classification Using K-NN," *Proc. Second Int. Conf. Electr. Syst. Technol. Inf. 2015 (ICESTI 2015) (pp. 75-81). Springer Singapore.*, pp. 75–81, 2016.
[28] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7–8, pp. 2031–2038, 2013.
[29] J. Zhang, X. Chen, S. Member, Y. Xiang, and S. Member, "Robust Network Traf fi c Classi fi cation," *IEEE/ACM Trans. Netw. (TON), 23(4), pp.1257-1270*, pp. 1–14, 2014.
[30] R. Lin, O. Li, Q. Li, and Y. Liu, "Unknown network protocol classification method based on semi-supervised learning," *Comput. Commun.*, pp. 300–308, 2015.
[31] W. Lu and L. Xue, "A Heuristic-Based Co-clustering Algorithm for the Internet Traffic Classification," *2014 28th Int. Conf. Adv. Inf. Netw. Appl. Work.*, no. 5, pp. 49–54, 2014.
[32] R. Alshammari and A. N. Zincir-Heywood, "Identification of VoIP encrypted traffic using a machine learning approach," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, no. 1, pp. 77–92, 2015.
[33] R. Krzanowski, "Burst (of packets) and burstiness," *66th IETF Meet.*, 2006.
[34] "Top Sites in United Kingdom - Alexa." [Online]. Available: https://www.alexa.com/topsites/countries/GB. [Accessed: 16-Feb-2018].

7. Acknowledgements

This research was undertaken with the support of my sponsor (Iraqi cultural attaché).

