Internet Traffic Classification Using Machine Learning

Tanjila Ahmed

Dec 6, 2017

UCDAVIS
UNIVERSITY OF CALIFORNIA

Agenda

1. Introduction
2. Motivation
3. Methodology
4. Results
5. Conclusion
6. References

Motivation

- Traffic classification is the categorizing of internet traffic according to various applications
- It is needed for network engineering, management, and control as well as other other analytics

[1]

Motivation

- Traditional techniques for traffic classification include port and payload based analysis.
- Encrypted data and dynamic port assignments make it harder to correctly identify the type of applications [2].
- Using a combinations of techniques for supervised and unsupervised learning algorithms has shown promise for classifying internet traffic [3].
- We wanted to explore how a K-mean clustering algorithm could be used to classify internet traffic.

Methodology Overview

- Capturing the trace of known applications
- Filtering out the noise to isolate the flow of interest
- Analyze patterns in the trace to grab all relevant features
- Feature Selection and clustering the data points
- Analyzing the error to determine the right number of clusters and fitting of the attributes
- Using those setting and data points as training data for another test set

K-Means Algorithm

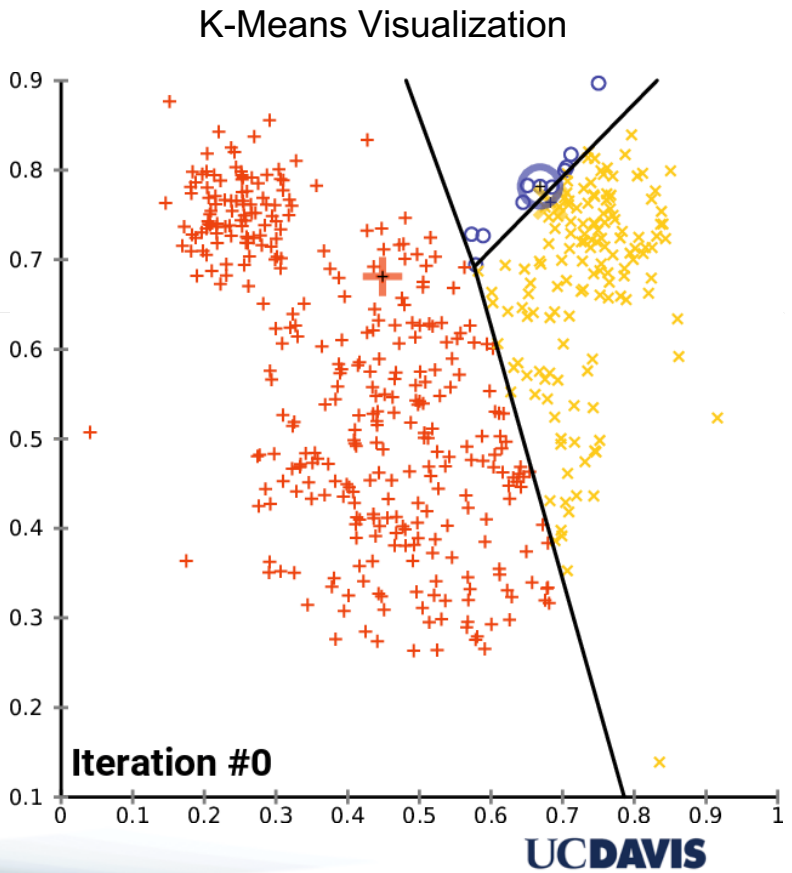
Unsupervised Learning Methods :

AutoClass [4], k-means [5] and DBSCAN [6]

K-mean : is the simplest to implement and least memory intensive [7]. Working process of it is very fast and robust [7]. Usually k-mean is widely used for both network anomaly detection [8,9] and traffic classification [6,10]. In [6] the authors made a qualitative comparison among above mentioned three unsupervised ML algorithms showing based on both accuracy and model building time k-mean gives the best solution for traffic classification.

K-Means Algorithm

- Implementation
  - Run the parsed data through the K-means algorithm to form clusters based on random positioning of the centroids
  - Iterate through the positions of the centroids until they converge
  - Determine the RMS error
- Analysis
  - Increase the number of centroids and until we see small changes in the RMS error.
  - This will give us an estimate of the correct number of groupings.



UCDAVIS

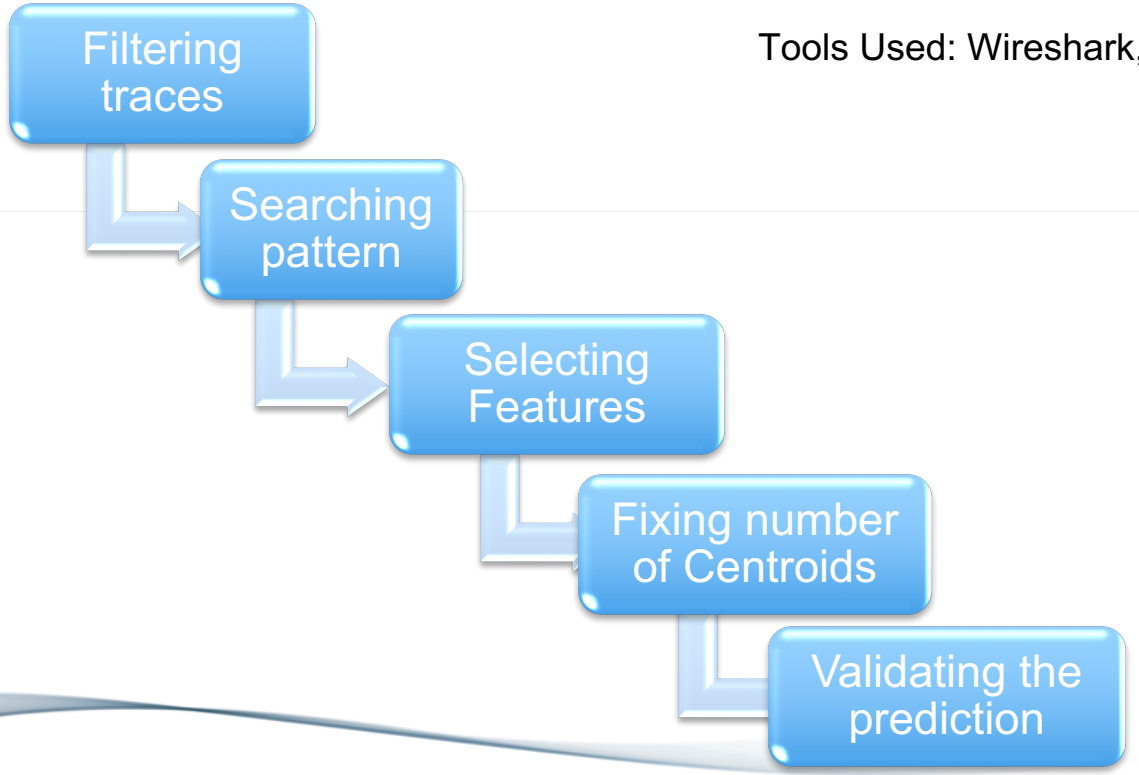
K-Means Algorithm

Using only k-mean algorithm is not sufficient for traffic classification. Although it can group the traffic based on their flow features but cannot identify the applications. Due to unavailability of labelled data and limitation of k-mean to identify applications we decided to go for a semi-supervised ML approach. Some of recent works [11,12] have shown some good progress using this semi-supervised approach. Semi-supervised approach uses fewer labeled data to predict traffic classes.

Training Set Vs Test Set

- By having a ground truth about the applications before running them through the K-means algorithm we can estimate the correct percentages in each cluster.
- We can’t know if these percentages correspond to the predicted application.
- So we use this data as a training set vs a test set of new known traces to see which cluster they fall into.
- This tell us type of application in each cluster

Result Generation Steps



Tools Used: Wireshark, Weka

Trace Filtering and Pattern Search

Filtering the traces meant removing everything other than specific application traffic in our wireshark capture.

1st approach :

- Applied a series of protocol filters to remove frames when wireshark was open but the target applications weren't running.
- Removal of 50- 75 % of all frames and made our trace incomplete

2nd Approach

- Using local ip and application port based filters we picked up more frames but might have lost few important connections
- Still more accurate representation of the traffic being generated by the application.

Feature Selection

- We had originally parsed out 9 features that were distinguishable between all the traces
  1. Number of Packets
  2. min/max/avg packet length
  3. duration
  4. min/max/avg inter-arrival time
  5. protocol
- Progressive selection
- With the applications we chose all 9 groupings seemed to give the best accuracy for the number of clusters.

Feature Selection

Table 1 : Percentage of actual groupings vs percentage of groupings with 1 -9 features added

Application	Original grouping	Grouping using features K=4	Grouping using features k=5
Online Games	33%	49%	41%
Facebook Messenger	31%	26%	29%
Youtube	21%	14%	15%
Download	13%	12%	9%
			7%

UCDAVIS

K-Centroids

- The number of centroids was chosen based on an approximation method called the **elbow graphing.**
- The number of centroids was increased from 2-7 to graphed vs Sum of Squared error in each cluster
- When the change noticeably slows down with the increase of centroids an elbow forms

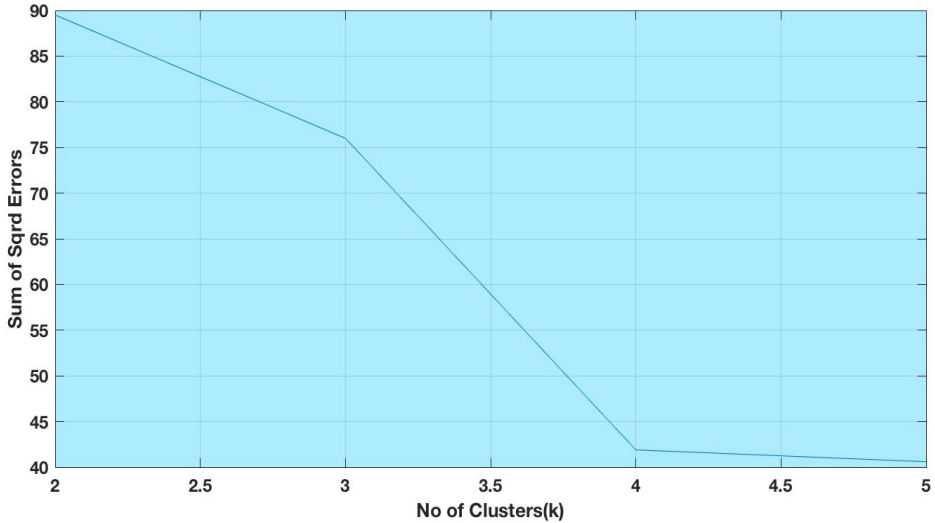


Fig. Elbow Graph
Approximation of the number of Centroids

UCDAVIS

Accuracy of Predictions

- We were able to correctly match 3 out of the 4 applications with a moderate rate of accuracy.
- “Download” traces were mis-grouped with Youtube and Facebook Messenger.
- This was most likely due to the filtering used in the trace and the type of download we were making.
- Similarities in used port (443 HTTPS) , protocol and long packet lengths.
- These were our some of the most weighted feature selections.

Accuracy of Predictions

Applications	Groups	Test 1 : Youtube	Test 2 : Online Games	Test 3: Facebook Messenger	Test 4: Download
Youtube	0	**64%**	14%	13%	42%
Download	1	11%	12%	0%	8%
Online Games	2	5%	**49%**	0%	9%
Facebook Messenger	3	20%	26%	**87%**	42%

Accuracy of Predictions

Applications	Actual Testset Grouping	Predicted Testset Grouping	Actual number of flows in testset	Predicted number of flows in testset
Youtube	20.3%	25%	81	100
Download	19.8%	10%	79	39
Online Games	31.1%	37%	124	147
Facebook Messenger	28.8%	28%	115	113

Conclusion

Factors needed to be carefully tuned for accurate internet traffic class predictions :

1. Large and diverse enough sample set to make sure you have enough data for groupings
2. Filtering the traces to remove the noise is crucial in order to form a reliable baseline for your training set
3. Tuning the feature selection and the number of K-centroids can have drastic effects on the resulting groupings

References

[1] EEC 274 lecture notes from Prof. Chen-neh Chuah.
[2] L. Yingqiu, L. Wei, L. Yunchun, 2007, Network Traffic Classification Using K-Means Clustering, Network Technology Key Lab of Beijing,
[3] Nguyen, Thuy TT, and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning." *IEEE Communications Surveys & Tutorials* 10, no. 4 (2008): 56-76.
[4] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in Proc. 2005 IEEE Conference on Local Computer Networks, pp. 250–257.
[5] J. Erman, A. Mahanti, and M. Arlitt, "Internet traffic identification using machine learning," in Proc. 2006 IEEE Global Telecommunications Conference, pp. 1–6.
[6] Erman, Jeffrey, Martin Arlitt, and Anirban Mahanti. "Traffic classification using clustering algorithms." In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pp. 281-286. ACM, 2006.
[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, pp. 50–58, Apr. 2010.
[8] R. Kumari, M. K. Singh, R. Jha, and N. K. Singh. "Anomaly detection in network traffic using K-mean clustering." In *Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on*, pp. 387-393. IEEE, 2016.
[9] Y. Shi, X. Peng, R. Li, and Y. Zhang. "Unsupervised Anomaly Detection for Network Flow Using Immune Network Based K-means Clustering." In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 386-399. Springer, Singapore, 2017.