# 5G Networks: Performance Visualization and Traffic Analysis

Biplav Choudhury
Virginia Tech
USA

Shashank Thakalapally
Virginia Tech
USA


**Figure 1.** Different applications enabled by 5G networks

## Abstract

While 5G is ushered to be the be the next big change in wireless communications, the gains in performance in real 5G deployments have yet to be quantified due to the relatively slow pace of 5G deployments. In this work, we use the world's first publicly available dataset that contains throughput, channel, and context information for a deployed 5G network to investigate this gain. While telecom companies promise significant improvements over the existing networks, this dataset allows us to to exactly compare 5G network performance in different scenarios. We will use visualization tools to help us to quantify the network performance, so that a comprehensive picture of a 5G network can be seen. Then we move into the traffic analysis of such a network using a different : 5G being a packet based network, will enable a range of different applications which has to be identified properly so that adequate resources can be assigned to it. We try to perform traffic classification on the traces to identify the applications to which the trace belongs to. We first use classification techniques and compare their performance. This dataset also contains traces of malware attacks in a very less proportion (0.05%), and we try to use some of the existing anomaly detection techniques to identify such malicious traffic. This work therefore tries to investigate some of the key issues regarding 5G deployment - performance gains and traffic classification.

***Keywords:*** 5G, network performance, visualization, classification, telecom quality of service

**ACM Reference Format:**
Biplav Choudhury and Shashank Thakalapally. 2018. 5G Networks: Performance Visualization and Traffic Analysis. In *Proceedings of CS5525 Final Course Project (CS 5525).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*CS 5525, December 10, 2020, Blacksburg, VA*

© 2018 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

## 1 Introduction

There has been a steady development in cellular communications from the first generation to the fourth generation (4G LTE) systems. The current bandwidth demands on the 4G network was never anticipated as these systems have evolved which have produced many different types of applications that were never thought of before. 5G has been heralded as the new era of communications where high data rates, very low latency, and extremely reliable communication is achieved. Some of the most promising applications that will be enabled via 5G are remote surgery, augmented/virtual reality, 4K high definition video streaming etc [19]. Keeping the requirements of such resource heavy applications, there have been several theoretical studies on the expected improvements over conventional networks that 5G is expected to bring to the user. A good survey of 5G networks is [11].

These studies usually consider idealized cases like ideal channel conditions [23], proper feedback [8], etc. Additionally, the effect of the hardware used is neglected as it can only be modelled with specific hardware. Therefore, to investigate the actual communication performance achieved with 5G, we analyse the data made available by Cork University [14] which claims to be the first publicly available data set that contains throughput, channel, and context information for 5G networks. As theoretical studies project 5G to reach rates up to 10 Gbps [21], it needs to be seen if such speeds will be experienced by the user. We also plan on classifying to identify which application is running at the user-end based on the quality-of-service metrics experienced by the user. These improvements are measured in metrics termed quality of service (QoS). However the absence of a standardized data set about the QoS offered/experienced has hindered an in depth study of the actual performance experienced by an user.

Moreover due to the expected increase in the QoS, there have been a host of new applications that will be running on 5G networks. 5G makes it possible for the network administrator to allocate resources based on the requirements, so that a user playing an online game or streaming high quality vido gets more network resource than the user just browsing the internet. This necessitates the ability to identify the application a traffic trace was generated from, and here we see the use of the classification algorithms that we have learnt in the class. The dataset used for this purpose is a collection of Software Defined Networking (SDN) traces that were collected using high performance computers using a full duplex Ethernet [20]. The traffic in it represents the full network usage of a 24 hour period where the target label is the network flow, or basically the application type. There are a total of 248 features. A few features are extracted indirectly using the header information in the TCP acknowledgement requests and tool called tcp-trace is used for this [4]. The target labels, i.e. the network flow can have 10 possible classes and they are shown in Table 1. The traffic traces are separated into 10 different data blocks where each trace represents 28 minutes of traffic.

The report has been arranged as follows - in Sec. 2, we describe the datasets. Then in Sec. 3, we describe the techniques used and also measure the performance attained. This section is a combination of model building and model evaluation. Finally, we discuss the results from a network perspective in Sec. 4 and conclude the paper with it.

## 2 Dataset Description

As mentioned before, we are using two different datasets and the first dataset is a collection of network traces collected with the objective of measuring network throughput in different conditions [14]. It consists of 5G network metrics such as network throughput, context-related, and channel-related information collected from a major Irish mobile operator. The metrics are generated using an Android application called G-NetTrack Pro, which is used for network monitoring. G-NetTrack pro can capture the network metrics and throughput information on a range of andriod devices without requiring any rooted privileges. The data set is composed of cellular key performance indicators on the client-side. This application was installed on Samsung Galaxy S10 which supports 5G and the network metrics are captured for two mobility patterns namely static and driving, while running three types of services namely file download, Netflix, and Amazon Prime, for each mobility pattern. The network metrics were captured in this way until 80GB of data was downloaded. The data set has 83 traces with a total duration of 3142 minutes. The metrics captured in the data set include:

- *Longitude and Latitude*: mobile device's GPS coordinates
- *Timestamp*: sample's timestamp
- *Operator name*: name of the cellular operator.(anonymised)
- *Velocity*: mobile device's velocity in km/h
- *NetworkMode*: mobile communication's standard (2G / 3G / 4G / 5G)
- *State*: download process's state, whether I(idle, not downloading ) or D(downloading).
- *SNR*: signal-to-noise ratio value.
- *DL_bitrate and UL_bitrate*: download/uplink rate measured at the device(application layer)
- *RSRQ*: Ratio of RSRP and received Signal Strength Indicator(RSSI). Signal strength/quality is measured across all resource elements(RE).
- *RSSI*: RSSI value. RSSI represents a received power including a serving cell and interference and noise from other sources. RSRQ, RSRP and RSSI are used for measuring cell strength/coverage and therefor cell selection.

- *RSRP*: RSRP value. RSRP represents an average power over cell-specific reference symbols carried inside distinct RE. RSRP is used for measuring cell signal strength / coverage and therefore cell selection (dBm).
- *CQI*: Channel Quality Indicator value of a mobile device. CQI indicates data rate that could be transmitted over a channel(highest MCS with a BLER probability less than 10%), as the function of UE's receiver characteristics and SINR. It is a feedback provided by UE to the eNB. eNB selects an appropriate modulation scheme and coding rate, based on UE's prediction of the channel.
- $ping_{avg}, ping_{loss}, ping_{max}, ping_{min}, ping_{std}$: statistics of ping(average, loss, max, min, standard deviation).

The second data set used in this project is provided by [10] and it consists of sets of Transport Control Protocol (TCP) packets flow between multiple clients and server in an institution. The data was collected by a high performance network monitor using a full duplex Ethernet [20], with a timestamp resolution of up to 35 nanoseconds. The data was sub-sampled into 10 separate data sets with each set belonging to different period of time taken from the 24 hours period when the data was collected. Each data set represents 28 minutes of traffic and consists of multiple objects where each object represents a single flow of TCP packets between the client and the server. Each object is also described by a group of features. It is observed from 2 that due to a variable density of traffic during each constant period, there are different number of flows in each data block. The information regarding list of data sets, number of flows in each data set and their duration are provided in Fig 2.

| Data-set | Start-time | End-time | Duration | Flows (Objects) |
|---|---|---|---|---|
| entry01 | 2003-Aug-20 00:34:21 | 2003-Aug-20 01:04:43 | 1821.8 | 24863 |
| entry02 | 2003-Aug-20 01:37:37 | 2003-Aug-20 02:05:54 | 1696.7 | 23801 |
| entry03 | 2003-Aug-20 02:45:19 | 2003-Aug-20 03:14:03 | 1724.1 | 22932 |
| entry04 | 2003-Aug-20 04:03:31 | 2003-Aug-20 04:33:15 | 1784.1 | 22285 |
| entry05 | 2003-Aug-20 04:39:10 | 2003-Aug-20 05:09:05 | 1794.9 | 21648 |
| entry06 | 2003-Aug-20 06:07:28 | 2003-Aug-20 06:35:06 | 1658.5 | 19384 |
| entry07 | 2003-Aug-20 09:42:17 | 2003-Aug-20 10:11:16 | 1739.2 | 55835 |
| entry08 | 2003-Aug-20 11:52:40 | 2003-Aug-20 12:20:26 | 1665.9 | 55494 |
| entry09 | 2003-Aug-20 13:45:37 | 2003-Aug-20 14:13:21 | 1664.5 | 66248 |
| entry10 | 2003-Aug-20 14:55:44 | 2003-Aug-20 15:22:37 | 1613.4 | 65036 |

**Figure 2.** Broad Statistics of each data set. [13]

The starting of the time period of each sample to be recorded, was randomly chosen to provide a wider sample of mixing across the day. Though each data sets approximately represent the same period of time, due to the variation in activity throughout the day, there is fluctuation in number of objects per data set. The data sets are represented as text files in which each line represents an object. One or more packet travelling between two IP addresses using appropriate protocols and pair of ports is defined as object in this case. Every packet has this tuple of information related to protocol and the pair of ports used. Only TCP protocol and TCP objects are considered in this data set. TCP has well-defined start and end of the objects which simplifies our work. The complexity in defining an object occurs when the setup is incomplete or the tear-down is abnormal.

**Table 1.** Network Flows in the Second Dataset

| Flow | Application | Proportion of Dataset |
|---|---|---|
| WWW | Web | 86.9% |
| Mail | SMTP | 7.5% |
| Bulk | FTP | 3.05% |
| Services | DNS | 0.556% |
| P2P | Torrent | 0.555% |
| Database | Orcale | 0.701% |
| Multimedia | VLC | 0.153% |
| Attack | Virus | 0.475% |
| Interactive | SSH | 0.029% |
| Games | Counter-Strike | 0.002% |

We find many desirable properties in complete TCP objects. They allows us to compute nearly all the discriminators provided in each data. For example, we can differentiate client from the server using complete TCP objects and thus allows us to reliably identify the server and client ports. A wide variety of features are provided in the data set to characterise objects. There are a total of 248 features and due to the large number of it, we do not describe each of them. Features are of both numeric and non numeric types, and some of the important features include traffic statistics are *source IP address, connection duration, packet size* etc.[13]

Simple statistics like inter-packet timings and packet length and information derived from transport protocol such as ACK and SYN counts, are all included in the data set. Also, this information is provided based on each direction individually(client to server or server to client) and on both directions. Packets and packet header-sizes are counted for many packet statistics. Significant features such as round trip time estimates,total number of re transmissions and size of TCP segments are derived using the header information in the TCP acknowledgement requests and a tool called tcptrace is used for this [4]. Each object is described using three modes:

- *interactive*: data packets travelling in both directions.
- *idle*: no packets between server and client for more than two seconds.
- *bulk*: data packets in one direction and only acknowledgements in the other.

The features of time spent by the object in the idle mode, the time a flow spends in the bulk mode, and the flow duration are provided. The target labels, i.e. the network flow can have 10 possible classes and they are shown in Table 1.

## 3 Model Building and Evaluation

In this section, we will analyze different aspects of the 5G network -

- Visualize network throughput in different contexts.
- Network traffic classification
- Detection of malicious traffic using anomaly detection techniques.

**3.0.1 Visualization.** For the visualization part, we will use the first dataset. First the data is visualized across different contexts of application and mobility on the type of throughput achieved. Context means the application and the mobility, and will be referred to as such in the following sections.

As a first step, we investigate one of the basic principles of capacity in wireless communications which is defined by the classic Shannon's theorem - higher Signal to Noise Ration (SNR) should translate to a higher channel capacity measured in terms of data transfer. So we first see the relationship between SNR and Channel Quality Index (CQI), and then between the CQI and downlink data rate. CQI is a measure of the SNR received by the user - as the Base Station (BS) cannot measure the SNR received at the destination, the receiver provides a feedback to the BS which is then used to select various transmission parameters at the BS. Theoretically, a better SNR should be understood as a better channel which will increase channel capacity. In Fig. 3, we see that while higher SNR means higher CQI(e.g. CQI for SNR = 30dB is 14), but even with very low SNR of -10dB, CQI measured to be very high. This situation is not desirable as it will lead to data rates that are not suitable with that SNR. In [15], this was shown to have occurred as a result of interference (high noise) at the pilots used by the BS and we suspect a similar situation had arose when this data was recorded. A CQI of greater than 10 is not at all feasible for such less SNR, which points to the possibility of faulty measuring technique or hardware impairments. While this is an vendor specific issue, the standard MATLAB implementation uses a CQI-SNR mapping where any SNR less than 0 dB can get mapped to a CQI of 1 or 2 [1]. Then in Fig. 4 we show how CQI translates to higher data rates. Usually, this is expected to be a non-monotonic relationship as works like [12] have shown but we see that this is not the case - peak rates are seen for SNR 10-12 dB rather than around 14-15 dB. Therefore for the basic analysis of SNR-CQI and CQI-rates, we find a few points that cannot be predicted by theoretical studies and only a real deployment can shed light on them.

Another aspect that we would like to visualize is the effect of the mobility in the data rate - as one of the feature is *velocity*, we will see how the mobility effects data rate. Fig. 5 shows the SNR and the data-rate for driving and static scenarios. As the figure shows, mobile users experience better SNRs which also translates to higher data rates. Though there is a famous result in wireless networks that shows

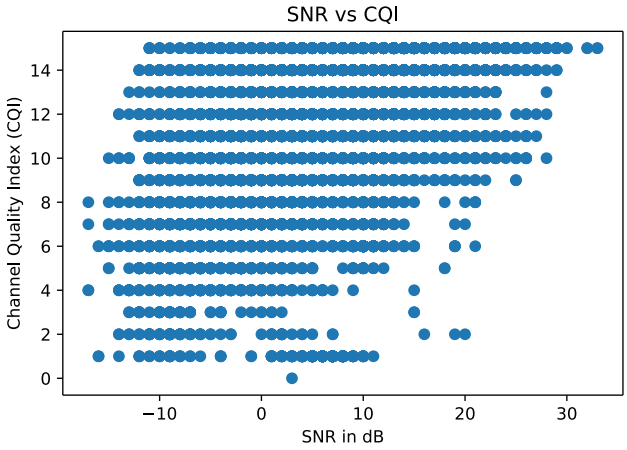


**Figure 3.** SNR vs CQI

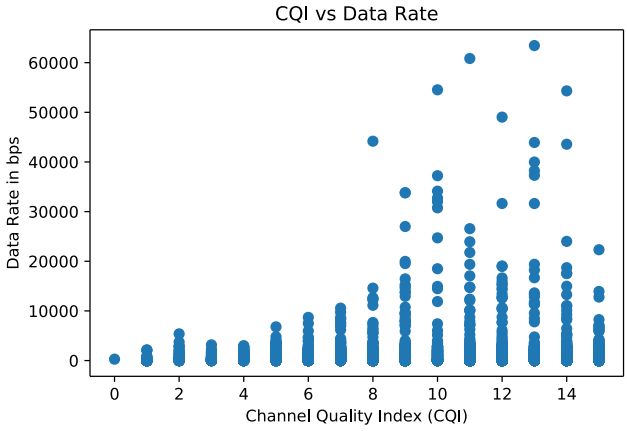


**Figure 4.** CQI vs Data

mobility improves connectivity [22] for ad-hoc networks, this is not what is happening here - in our case, only the BS transmits to the users so there is no ad-hoc network. We suspect the higher SNR could be a result of the users driving around regions with better coverage. We have the locations of the users but due to the absence of the location of the BS, we cannot corroborate our claim of mobile users passing through regions with better coverage.

The final aspect of 5G that we would like to explore is the variation in the rates depending on the service provided. The dataset has 3 types of service :

- Amazon Prime : a fixed amount of streaming video using Amazon Prime service.
- Netflix : a fixed amount of streaming video using Netflix service.
- File Downloading : download a 20 GB file from the nearest available server.

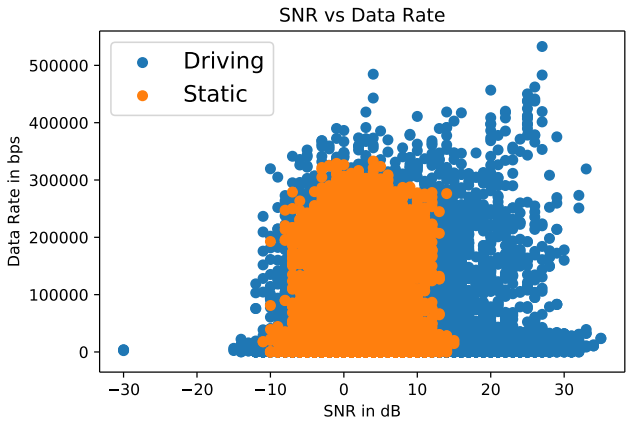


**Figure 5.** Effect of mobility on Data Rate

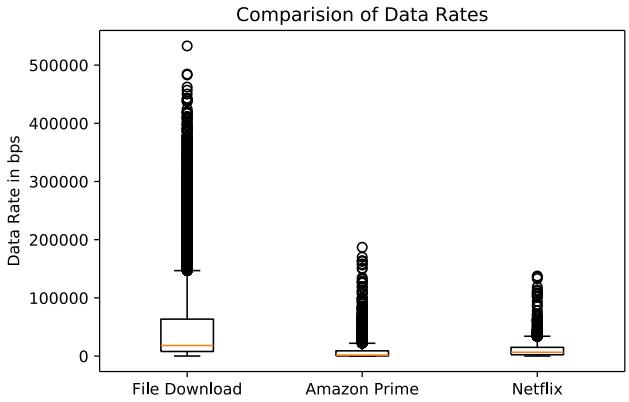


**Figure 6.** Download rates for different services

In this case, the performance depends on the service provider in addition to the network coverage. So if Amazon has a better content delivery network (CDN) [5], it will see the best rates. We use a box-plot to visualize the rates seen across the three services in Fig. 6 - we see that while the average rate denoted by the yellow line inside the box is similar for the three services, downloading offers the highest *peak* rates compared to the streaming applications. Additionally among the streaming applications, Amazon Prime offers better rates. This means is that the average performance experienced by an user is mostly the same as evident from the similar average across the three applications, and the higher data rates being mostly outliers for all the three applications.

**3.0.2 Classification.** In this section, we will be using the second dataset mentioned in Sec. 2 that contains traces of variety of applications running over an ethernet. Our aim is to be able to classify the different types of traffic to their respective applications. There is a new concept in 5G called the *network slicing* [6] where the BS can slice up the network into smaller parts based on the requirements of the users. However to do this, the network has to be able to identify the type of application that is generating the traffic and this is an area where we see a scope of various supervised classification techniques.

Before going into the model building, we need to perform some pre-processing steps on the data before we can use them for classification. The data we use consists of values ranging from less than zero to greater than 1000. This will create some bias as the higher values will dominate the smaller values. For all the values to be considered equally, we need to scale the data. Standard procedure for this is to find out the mean value of each feature and subtract the mean from every value of the feature. This will set the mean of the feature values to zero. Also, we find the standard deviation of the values of each feature and divide the feature values by it. This will set the variance of the feature values to 1. It is suggested to standardize the data set's features onto unit scale which is a requirement for the optimal performance of many machine learning algorithms.

At first, we use PCA to collect principal components that account for 95% variance. We use Principal Component Analysis (PCA) to reduce the dimensionality of the original feature space by projecting it onto a smaller subspace. Each value of the covariance matrix represents the covariance between two features and the covariance matrix is calculated for all the features. Then we perform eigen decomposition on the covariance matrix and find the eigenvalues and eigen-vectors. The eigenvalues and vectors determine the magnitude and the direction of the new feature space. We then order the eigenvalues in decreasing order and choose the top $n$ eigen values which contain the most information. In order to determine how many top $n$ eigen values to use, we use a measure called explained variance. Explained variance of eigen value is defined as the magnitude of an eigen value divided by the sum of all eigen value's magnitudes. This measure shows us how much variance each value is capturing. The first $n$ features which capture 95 percent of the variance are used for the classification and the rest of them are not considered and the result of the PCA analysis in terms of explained variance can be seen in Fig. 8.

For implementing the PCA in python, we combine the 10 sub datasets to a pickle file and use this pickle file to load the data into a variable. We see that the loaded data is of size 377526 x 249. We first remove all the features having any missing values and convert string features to categorical features. 216 features remain and this are considered for the future calculations. Fig. 8 shows the variance captured per number of components. We observe that 75 components capture 95 percent of variance. We start with a decision tree where cross validation was done for 5 folds and the resulting confusion matrix over all the classes is shown in Fig. 7. While

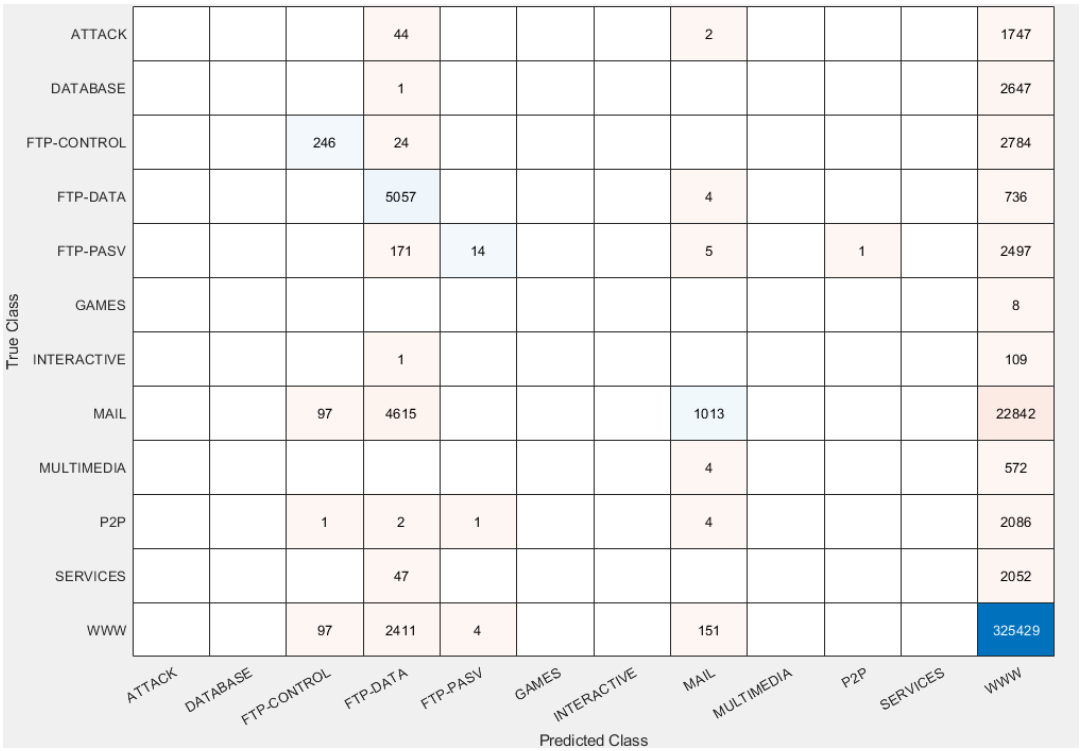


**Figure 7.** Confusion Matrix for a Tree Based classifier using PCA

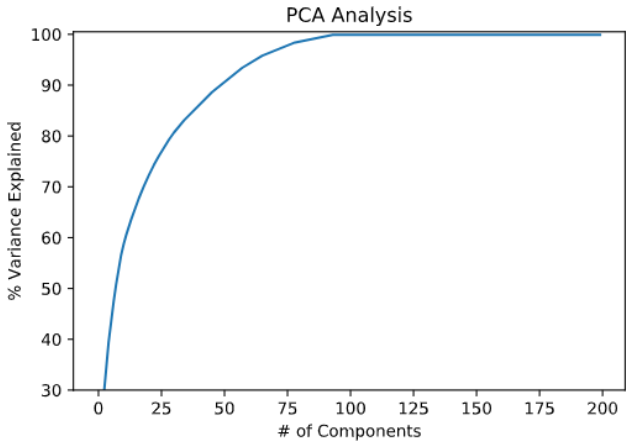


**Figure 8.** PCA Analysis showing variance captured per number of components

we get an accuracy of 87.9% using a decision tree, this should not be seen as an indication of a good performance : from Table 1, we know that almost 87% of the data belongs to a single type of traffic due to which any dummy classifier predicting each traffic sample to be that type will have at least 87% accuracy. We also tried other classifiers and the results are listed in Table 2.

- Support Vector Machine (SVM): SVM is a discriminative classifier which is defined by a separating classifier. It is a supervised learning model which outputs an optimal hyper plane to categorize new examples, for a given labeled training data. For example, the hyper plane in case of a 2D space will be a line dividing the plane into two parts with each class belonging to one part.
- Bagged Trees: Applying Bootstrap procedure to a high-variance machine learning algorithm such as decision trees, is called as Bagging. Bagging is a very powerful ensemble method which makes accurate predictions than individual models by combining predictions of independent base learners. It can be used to reduce the variance on algorithms having high variance.
- Boosted Trees: Achieving a strong learning algorithms by combining many sequentially connected weak learners in series, is called boosting. In boosted trees, the weaker learners are decision trees. The weaker learners are combined in series such that each tree attempts to minimize the error of previous trees and therefore make boosting a accurate and highly efficient model. Generally boosting is seen as a better alternative to boosting. We will use two versions of boosting - AdaBoost [16], the most popular implementation and RUSBoost [17], which is supposed to be better suited for imbalanced datasets like ours.

**Table 2.** Classifier Performance

| Classifier | Accuracy |
|---|---|
| Naive Bayes (NB) | 86.9% |
| Support Vector Machine (SVM) | 7.5% |
| Bagged Trees | 87.8% |
| Ada-Boosted Trees | 87.8% |
| RUS-Boosted Trees | 65.5% |
| Neural Network | 86.93% |

- Neural Network: We also decided to test a simple neural network of 2 hidden layers with 512 and 256 neurons in them. The testing of the full dataset over an adequate number of epochs took a very long time, therefore we made a smaller sample of the dataset and tested it only over 500 epochs to get a sense of how the neural network behaves. Softmax and Recified Linear Unit activation functions were used at each of the layers.

As expected, NB does very bad and this can be attributed to its inherent assumption of all features being independent : in network traffic, features like application, incoming/outgoing port, connection protocol, packet size etc. are all related and therefore assuming them to be independent will result in mistakes. SVM give an accuracy of around 80.7%. Then we try the two types of ensemble classifiers taught in the class, bagging and boosting, with the base classifiers being trees which resulted in accuracies as shown in Table 2. Some important observations that we made in this step are -

- Using all the features (200+) takes a long time to train, and using dimensionality reduction techniques like PCA had an added benefit of reducing this time by focusing on the principal components that contribute the most to its variance. For example in our case, we are using around 75 principal components rather than all 200 plus raw features.
- Even though the performance of RUS Boosted Trees is worse compared to AdaBoost Trees in terms of accuracy (see Table 2), RUS Boosted Trees does better in terms of identifying some of the minority classes like ATTACK and DATABASE. This can be seen in the ROC-AUC curve of the two as shown below in Fig. 9 and 10 where we treat the ATTACK class as positive and all others as negative class : RUS-Boosted trees has an AOC of 0.79 while Ada-Boosted trees has an AOC of 0.7 in classifying ATTACK traffic.

Coming back to the tree based classifier, it can be also seen from Fig. 7, a lot of traffic classes were not identified at all. One of this traffic is the type ATTACK, which points to malicious virus and worms. This could be either of two reasons -

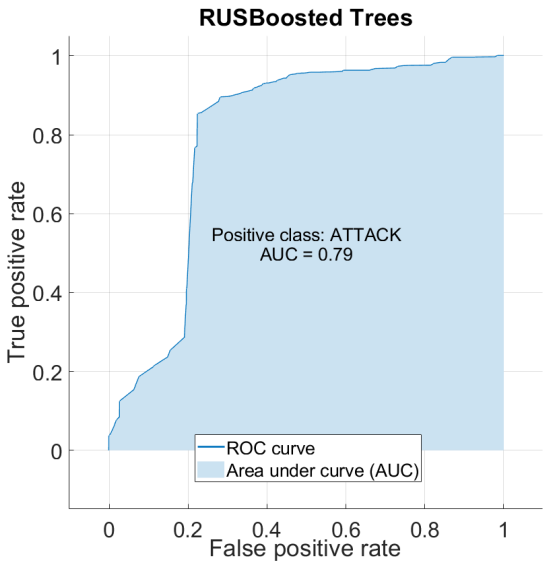


**Figure 9.** ROC curve for RUS-Boost

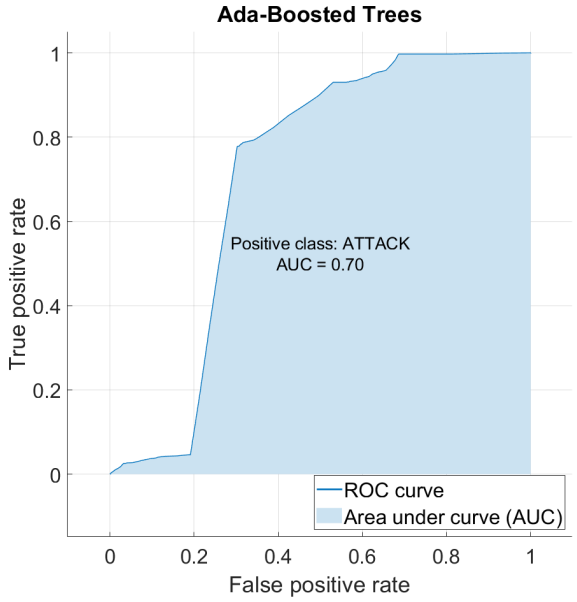


**Figure 10.** ROC curve for AdaBoost

- ATTACK only forms 0.475% (1793 samples) of the data so the classifier doesn't have a lot of samples to learn about it.
- The virus/worm designer created it so that it could not be distinguished from other type of traffic.

So in the next section, we will focus on trying out anomaly detection techniques to identify these ATTACK type traffic.

**3.0.3 Anomaly Detection.** As mentioned in the previous section, around 0.475% of the data is classified as ATTACK

and this presents a nice use case for anomaly detection technique. For this case, we modify the data to be a binary class label where the attack traffic numbering 1793 samples is classified as 1 while all other traffic are 0. We employ two techniques for this step -

- Isolation Forest : The isolation forest algorithm is based on the idea of recursively generating partitions on the sample by selecting one of the attributes randomly and a split value for it. It operates in two steps. First, an ensemble of isolation trees (iTrees) is built from the training dataset. Then each sample of the test data is passed through the iTrees and an anomaly score assigned to them with the logic that an higher anomaly score means there is a higher chance of the sample to be an anomaly. After assigning such a score to each of the points, a threshold is selected so that all points with a score greater than this threshold is identified as an anomaly [2].
- Local Outlier Factor - This method is based on the concept of “local density" where locality refers to a fixed number of neighbors, a tunable parameter. The distance to these neighbors then determine the density. Similar points have similar densities, and therefore anomalous points can be identified by finding points with abnormal densities [3]. For the simulations, we keep the number of neighbors as 2.

| | Predicted | |
|---|---|---|
| Actual | 119 | 1674 |
| | 37634 | 338099 |

**Figure 11.** Confusion Matrix for Isolation Forest

| | Predicted | |
|---|---|---|
| Actual | 188 | 1605 |
| | 37565 | 338168 |

**Figure 12.** Confusion Matrix for Local Outlier Factor

The confusion matrices as a result of two methods above are shown below in Fig. 11 and 12 respectively. As it can be seen, Isolation Forest is able to identify 119 of the malicious traffic while Local Outlifier Factor is able to identify around 188 of the malicious traffic, which means the precision is around 6% and 10% respectively. As is the case with imbalanced datasets, the true-negatives are very high as majority of the data is not anomalous. There might be some room for improvement in this case with parameter optimization, but we were unable to find one. Therefore we only get a slight improvement in our ATTACK traffic identification from our tree based classifier presented in Fig. 7. This presents a strong case for advanced anomaly detection techniques in the context of network traffic, and in [10], the authors do a forward subset selection where they find a set of 13 features that best represent the data. Using these 13 features, the authors were able to get around 13% precision in identifying the anomalous traffic.

## 4 Discussion

In this work, we took a two step approach of investigating different aspects of 5G networks. In the first step, we focus on finding out the quality of service experienced by different users. We see significant deviation from the expected results from theoretical studies - we find that the data rates achieved by an user may not be a monotonic function of the channel quality index or the signal strength. While the underlying cause of this phenomenon is difficult to identify, we can speculate that this was caused by interference in the pilot signals or issues with the hardware used. We also see that mobile users experience better data rates, and this can be attributed to the fact that mobile users are more likely to move into regions with better coverage. While we cannot verify this claim too, it is widely accepted that the channel fading effects is very stong in residential places where there are lots of obstructions to line of sight propagation of signals and this could be a reason why static users are not getting better rates. Some of the proposed solutions are femto cells [7], close rage mmWave antennas [9] etc.

Then the second part of the work is focused on the fact that 5G networks have to be able to identify the application generating the traffic because it needs to dynamically create network slices and allocate resources to it. We compare different classifiers and see that the performance of most of them are very similar, and this might be a common scenario for datasets where a single type of data dominates - in our case, usual internet traffic constitutes around 87% of the data thereby making any accuracy performance of around 87% insignificant. We also tried a deep learning approach for the classification - we used google collab but the time taken was very long due to which we could not use the full dataset. When using smaller samples of the data, the deep learning network of 2 hidden layers didn’t provide any significant improvement over the previous approaches we tried. An interesting scenario was observed between AdaBoost and RUSBoost - while AdaBoost had an overall better accuracy, RUS boost was able to identify certain minority classes with better precision. This highlights the importance of appropriate performance metrics for imbalanced datasets.

We then finally come to anomaly detection where we focus on identify malicious traces of traffic. This is an important aspect for 5G as with more information passing through the networks, more concern is being raised on privacy and other forms of attacks. We use two algorithms for the anomaly detection and see that they offer precision of less than 10%, which means that a lot of these traffic could go unnoticed. This might also be a result of the dataset not containing all

the appropriate features that might be needed for a realistic traffic classification task. Recent advances in deep learning have shown its power in anomaly detection for wireless traffic too [18], and it would be very interesting to see its applicability in our dataset.

## References


[1] [n.d.]. 5G NR CQI Reporting - MATLAB & Simulink. https://www.mathworks.com/help/5g/ug/5g-nr-cqi-reporting.html. (Accessed on 12/10/2020).

[2] [n.d.]. Isolation forest - Wikipedia. https://en.wikipedia.org/wiki/Isolation_forest. (Accessed on 12/07/2020).

[3] [n.d.]. Local outlier factor - Wikipedia. https://en.wikipedia.org/wiki/Local_outlier_factor. (Accessed on 12/07/2020).

[4] [n.d.]. TCP Traceroute. https://catonmat.net/tcp-traceroute. (Accessed on 12/07/2020).

[5] [n.d.]. What is a CDN? | How do CDNs work? | Cloudflare. https://www.cloudflare.com/learning/cdn/what-is-a-cdn/. (Accessed on 11/29/2020).

[6] NGMN Alliance. 2016. Description of network slicing concept. *NGMN 5G P* 1, 1 (2016).

[7] Jeffrey G Andrews, Holger Claussen, Mischa Dohler, Sundeep Rangan, and Mark C Reed. 2012. Femtocells: Past, present, and future. *IEEE Journal on Selected Areas in communications* 30, 3 (2012), 497–508.

[8] Gilberto Berardinelli, Saeed R Khosravirad, Klaus I Pedersen, Frank Frederiksen, and Preben Mogensen. 2016. Enabling early HARQ feedback in 5G networks. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.

[9] Laurent Dussopt, Ossama El Bouayadi, Jose Alberto Zevallos Luna, Cédric Dehos, and Yann Lamy. 2015. Millimeter-wave antennas for radio access and backhaul in 5G heterogeneous mobile networks. In *2015 9th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 1–4.

[10] Zhong Fan and Ran Liu. 2017. Investigation of machine learning based network traffic classification. In *2017 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 1–6.

[11] Akhil Gupta and Rakesh Kumar Jha. 2015. A survey of 5G network: Architecture and emerging technologies. *IEEE access* 3 (2015), 1206–1232.

[12] Mohammad T Kawser, Nafiz Imtiaz Bin Hamid, Md Nayeemul Hasan, M Shah Alam, and M Musfiqur Rahman. 2012. Downlink snr to cqi mapping for different multipleantenna techniques in lte. *International journal of information and electronics engineering* 2, 5 (2012), 757.

[13] Andrew Moore, Michael Crogan, Andrew W. Moore, Queen Mary, Denis Zuev, Denis Zuev, and Michael L. Crogan. 2005. *Discriminators for use in flow-based classification.* Technical Report.

[14] Darijo Raca, Dylan Leahy, Cormac J Sreenan, and Jason J Quinlan. 2020. Beyond throughput, the next generation: a 5G dataset with channel and context metrics. In *Proceedings of the 11th ACM Multimedia Systems Conference*. 303–308.

[15] Raghunandan M Rao, Vuk Marojevic, and Jeffrey H Reed. 2019. Analysis of non-pilot interference on link adaptation and latency in cellular networks. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 1–5.

[16] Robert E Schapire. 2013. Explaining adaboost. In *Empirical inference*. Springer, 37–52.

[17] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2009. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40, 1 (2009), 185–197.

[18] Vrizlynn LL Thing. 2017. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[19] Hanif Ullah, Nithya Gopalakrishnan Nair, Adrian Moore, Chris Nugent, Paul Muschamp, and Maria Cuevas. 2019. 5G communication: an overview of vehicle-to-everything, drones, and healthcare use-cases. *IEEE Access* 7 (2019), 37251–37268.

[20] Pan Wang, Feng Ye, Xuejiao Chen, and Yi Qian. 2018. Datanet: Deep learning based encrypted network traffic classification in sdn home gateway. *IEEE Access* 6 (2018), 55380–55391.

[21] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, and Wenbo Wang. 2017. A survey on mobile edge networks: Convergence of computing, caching and communications. *Ieee Access* 5 (2017), 6757–6779.

[22] Xinbing Wang, Xiaojun Lin, Qingsi Wang, and Wentao Luan. 2012. Mobility increases the connectivity of wireless networks. *IEEE/ACM transactions on networking* 21, 2 (2012), 440–454.

[23] Lei Zhang, Ayesha Ijaz, Pei Xiao, Mehdi M Molu, and Rahim Tafazolli. 2017. Filtered OFDM systems, algorithms, and performance analysis for 5G and beyond. *IEEE Transactions on Communications* 66, 3 (2017), 1205–1218.