WiSig
Networks

Procedure Document

“5G BS Protocol Stack”

Table of Contents

Table of Figures

No table of figures entries found.

1 Introduction

Accurate traffic classification is of fundamental importance to network activities, like Quality of Service to providing operators with useful forecasts for long-term provisioning. We apply a Naive Bayes estimator to categorize traffic by application. Uniquely, the work capitalizes on hand-classified network data, using it as input to a supervised Naiıve Bayes estimator. In this document we illustrate the high level of accuracy achievable with the Naive Bayes estimator. We further illustrate the improved accuracy of refined variants of this estimator. Experimental results indicate that with the simplest of Naive Bayes estimator they were able to achieve about 65% accuracy on per-flow classification and with two powerful refinements they have improved this value to better than 95%; this is a vast improvement over traditional techniques that achieve 50–70%. While their technique uses training data, with categories derived from packet-content, all of their training and testing was done using header-derived discriminators.This is emphasized as a powerful aspect of their approach: using samples of well-known traffic to allow the categorization of traffic using commonly-available information alone.

1.1 Definitions, Acronyms, and Abbreviations

MAC	: Media Access Control
QOS	: Quality of Service
PHY	: Physical Layer
FPGA	: Field Programmable Gate Array
FAPI	: Femto Application Platform Interface

NR : New Radio

1.2 References

[1] D. Moore, K. Keys, R. Koga, E. Lagache, and K. C. Claffy. CoralReef software suite as a tool for system and network administrators. In Proceedings of the LISA 2001 15th Systems Administration Conference, December 2001.

[2] C. Logg and L. Cottrell. Characterization of the Traffic between SLAC and the Internet, July 2003. http://www.slac.stanford.edu/comp/net/slacnetflow/html/SLAC-netflow.html.

[3] A. W. Moore and D. Papagiannaki. Toward the Accurate Identification of Network Applications. In Proceedings of the Sixth Passive and Active Measurement Workshop (PAM 2005), March 2005.

[4] T. Karagiannis, A. Broido, M. Faloutsos, and k c claffy. Transport layer identification of P2P traffic. In Proceedings of Internet Measurement Conference, Taormina, Sicily, Italy, October 2004.

[5] A. W. Moore. Discrete content-based classification — a data set. Technical report, Intel Research, Cambridge, 2005.

[6] V. Paxson. Empirically derived analytic models of wide-area tcp connections. IEEE/ACM Trans. Netw., 2(4):316–336, 1994.

[7] K. C. Claffy. Internet traffic characterization. PhD thesis, University of California, San Diego, 1994.

[8] Christian Dewes, Arne Wichmann, and Anja Feldmann. An analysis of internet chat systems. In IMC ’03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pages 51–64, 2003.

[9] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. IEEE/ACM Trans. Netw., 3(3):226–244, 1995.

[10] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: A statistical signature-based approach to IP traffic classification. In ACM SIGCOMM Internet Measurement Conference, Taormina, Sicily, Italy, 2004

2 Overview

Accurate network traffic classification is fundamental to network activities, Quality of Service to providing operators with useful forecasts for long-term provisioning. Yet, classification schemes are difficult to operate correctly because the knowledge commonly available to the network, i.e. packet-headers, often does not contain sufficient information to allow for an accurate methodology. This leads to traditional techniques for traffic/flow classification that are often no-more accurate than 50–70%.The method discussed in this document uses supervised Machine-Learning to classify network traffic. Uniquely, we use data that has been hand-classified to one of a number of categories. Sets of data consisting of the category combined with descriptions of the classified flows are used to train the classifier. We test our algorithm using data-sets consisting of only the object descriptions and then compare the predicted category with the actual category for each object. In the process of applying Naive Bayes we plan to provide insight into the behavior of this technique itself. We will illustrate the sensitivity of the Naive algorithm to its initial assumptions and we plan to demonstrate that the use of two techniques, one to break the Gaussian assumptions and the other to improve the quality of discriminators as input, lead to significant improvements in the accuracy of the Naive Bayes technique.

3 Procedure

3.1 Algorithms used

3.1.1 Bayes Theorem

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)}$$

where:

$P(A)$= The probability of A occurring

$P(B)$= The probability of B occurring

$P(A|B)$=The probability of A given B

$P(B|A)$= The probability of B given A

$P(A \cap B))$= The probability of both A and B occurring

3.1.2 Naive Bayes Assumption

Consider a data sample x = (x1, . . . , xn) which is a realization of X = {X1, . . . , Xn} such that each random variable Xi is described by m attributes {A1, . . . , Am} (referred to as discriminators) that can take numeric or discrete values. Xi = A (i) 1 , ... , A (i) m T is then a random vector. As an example, for Internet traffic, A (i) j may represent the mean inter-arrival time of packets in the flow i. Assume now that there are k known classes of interest. Let C = {c1, . . . , ck} represent the set of all known classes. For each observed instance xi in x, there is a known mapping C : x → C representing the membership of instance xi to a particular class of interest. The notation C(xi) = cj stands for "the instance xi belongs to the class cj ". Bayesian statistical conclusions about the class cj of an unobserved flow y are based on probability conditional on observing the flow y. This is called the posterior probability and is denoted by p(cj | y). The celebrated Bayes rules gives a way of calculating this value: p(cj | y) = p(cj)f(y | cj) /p(cj)f(y | cj) where p(cj) denotes the probability of obtaining class cj independently of the observed data, f(y | cj) is the distribution function and the denominator acts as a normalizing constant. The goal of the supervised Bayes classification problem is to estimate f(y | cj), j = 1, . . . , k given some training set x. To do that, Naive Bayes makes certain assumptions on f(· | cj) such as independence of Ai's and the standard Gaussian behavior of them. The problem is then reduced to simply estimating the parameters of the Gaussian distribution and the prior probabilities of cj 's. In fact, Naive Bayes is also capable of dealing with discrete random discriminators, which could represent the state of some flag of a flow, by treating them independently and using the frequencies of occurrence to estimate f(· | cj), j = 1, . . . , k. In reality, independence and normality assumptions are flawed for the problem in consideration:

- Different discriminators are clearly not independent. An example is packet-header size statistics. The TCP header size may vary directly in proportion with the total packet-length no matter what the other characteristics of the traffic might be.

- Assumption of the normality of discriminators is also inaccurate. Notable problems arise when the real distribution is multimodal.

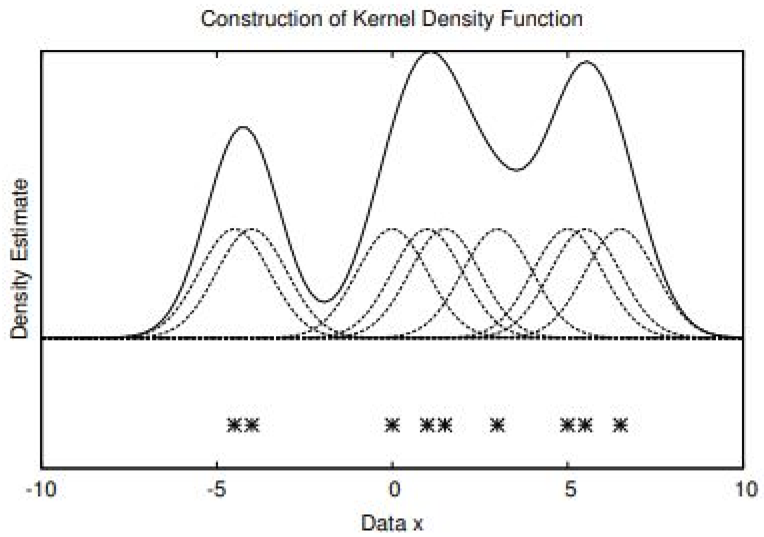
3.1.2 Naive Bayes Kernel Estimation Function

The Naive Bayes Kernel estimation method is a generalization of Naive Bayes and it addresses the problem of approximating every discriminator by a Gaussian distribution as described in the previous section.

Naive Bayes Kernel Estimation is similar to Naive Bayes method algorithmically and the only difference arises in estimating f(· | cj), j = 1, . . . , k. Whereas Naive Bayes estimated f(· | cj), j = 1, . . . , k by fitting a Gaussian distribution over the data, Kernel Estimation, as the name suggests, uses kernel estimation methods, i.e. the estimate of the real density f(· | cj) is given by

$$\hat{f}(t \mid c_j) = \frac{1}{n_{c_j} h} \sum_{x_i : C(x_i) = c_j} K\left(\frac{t - x_i}{h}\right),$$

where h is called the kernel bandwidth and K(t) is any kernel, where kernel is defined as any non-negative function. Naive Bayes kernel estimation procedure will use Gaussian density as the kernel for the analysis, party because it is Gaussian density has desirable smoothness properties. Figure 1 illustrates how an estimate of the density f(·) is constructed. Assume that some sample of data points has been collected. These points are represented by crosses on the x-axis. For each data point a Gaussian distribution centered on this point is fitted. A summation of those functions gives an estimate of the real density of the data distribution.



3.1.3 Dimensionality reduction using FCBF

Discriminator selection and dimension reduction plays a very important role in Machine Learning by acting as a preprocessing step, removing redundant and irrelevant discriminators. The prediction accuracy of the Naive Bayes algorithm suffers from the presence of irrelevant and redundant attributes. The ability to identify the most important discriminators of the Internet traffic is useful not only because the results will reveal what discriminators are best for traffic classification, but also because classification accuracy can be improved and, a reduction in the number of flow discriminators is computationally attractive.

It is important to define what is meant by irrelevant and redundant discriminators. Definition 1: A discriminator is said to be irrelevant if it carries no information about different classes of interest. This discriminator has no discriminative power. For example, a discriminator may take only a single value and, on that basis, no classification could be made.

Definition 2: A discriminator is said to be redundant if it is highly correlated with another discriminator. The reason for removing redundant discriminators is supported by the fact that redundant discriminators either worsen the accuracy or increase over-fitting

There exist two different approaches to discriminator selection, namely the filter and the wrapper methods. Filter methods use the characteristics of the training data to determine the relevance and importance of certain discriminators to the classification problem. For example, a measure of relevance could be the degree of correlation between discriminators and the class, or some measure of separation of classes based on the discriminator in consideration. On the other hand, wrapper methods make use of the results of a particular classifier (e.g. Na¨ıve Bayes) to build the optimal set by evaluating the results of the classifier on a test set for different combinations of discriminators. By reiterating the algorithm, the user can identify the optimal set of discriminators that are suited for a particular classification method (e.g. Na¨ıve Bayes). One such method would be the “forward selection” method, where one starts with no attributes and progressively adds one attribute after the other, checking the outcome after each additional attribute. A second example is “backward elimination”, where one starts with the full set of discriminators to work “backward” by eliminating discriminators one after the other. The main drawback of this method is that it is very computationally expensive, particularly in high dimensions as one needs to try each combination of a number of different discriminators.

In this process we use FCBF: Fast Correlation-Based Filter (FCBF), as well as a variation of a wrapper method in determining the value of the threshold . The FCBF filter method performs very well in improving the performance of Naiıve Bayes when contrasted with other related techniques . A discriminator is good if it is relevant to the class concept but it is not redundant to any of the other relevant features. In FCBF, goodness of a discriminator is measured by its correlation with the class and other good attributes. That attribute becomes good if it is highly correlated with the class, yet not correlated with any other good attributes. The correlation measure used in FCBF is based on the entropy of a random variable — a measure of uncertainty. The entropy of a discrete random variable X taking values in {x1, . . . , xn} is given by and for two variables is given by

$$H(X) = -\sum_{x_i} p(x_i) \log_2 p(x_i),$$

$$H(X \mid Y) = -\sum_{y_j} p(y_j) \sum_{x_i} p(x_i \mid y_j) \log_2 p(x_i \mid y_j),$$

information gain is given by

$$IG(X \mid Y) = H(X) - H(X \mid Y).$$

using above equations we define symmetric uncertainty which is given by

$$SU(X, Y) = 2 \left[\frac{IG(X \mid Y)}{H(X) + H(Y)} \right]$$

Symmetrical uncertainty takes values in [0, 1], where the value 1 means that the knowledge of discriminator values induces the value of the other, while 0 suggests that attributes X and Y are independent. By this point, Equation has only been defined for nominal values 6 , therefore FCBF continuous discriminators discrete before the core analysis .

The FCFB algorithm selects good discriminators via a two stage process by:

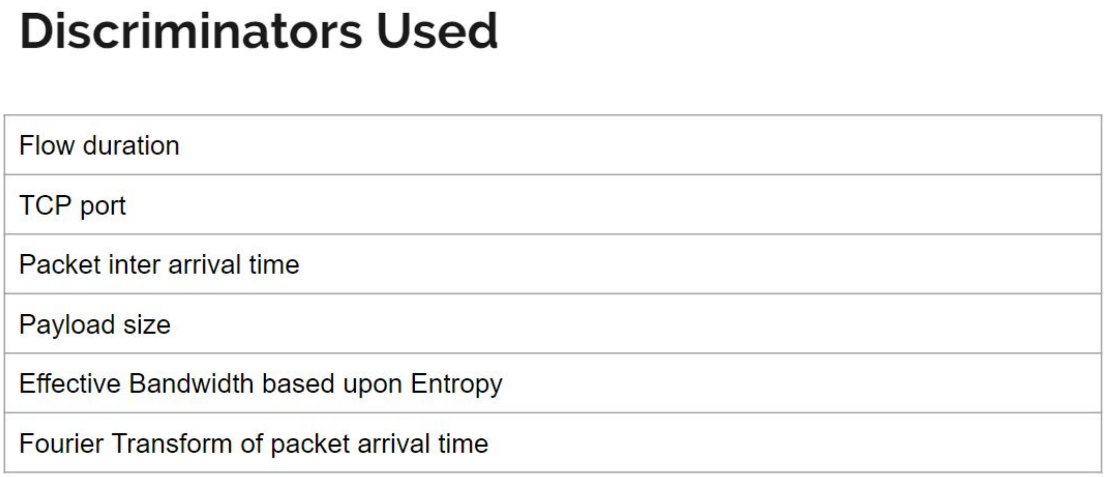
- identifying the relevance of a discriminator
- identifying the redundancy of a feature with respect to other discriminators.

The following algorithm is used to identify the best number of discriminators to be used for a particular training set.

- Rank all discriminators in order of importance as calculated by the FCBF method.
- The goal is to determine how many of the most important discriminators to choose. To do that an independent set of test data is chosen and it is used to evaluate the performance of Na ¨ıve Bayes classifier trained on different number of discriminators.
- For n from 1 to m, train Na ¨ıve Bayes on the training set with n discriminators and evaluate the resulting classifier on the test set.

3.2 Object and Discriminators

The application of a classification scheme requires the parameterizations of the objects to be classified. Using these parameters the classifier allocates an object to a class. Due to their ability to allow discrimination between classes, we refer to these object-describing parameters as discriminators. The fundamental object classified in our approach is a traffic-flow which is represented as a flow of one or more packets between a given pair of hosts. The flow is defined by a tuple consisting of the IP address of the pair of hosts, the protocol type (e.g., ICMP, TCP or UDP) and, in the case of UDP and TCP, the port numbers used by the two hosts. In the case of TCP, a flow has a finite duration defined by the semantics of the TCP protocol. For our work we limit ourselves to training and testing sets that consist only of TCP and are made-up of semantically 2 complete TCP connections. Our use of only complete TCP flows is a simplification that allows us to concentrate upon the classification process — rather than describe the mapping of datagram and partial-flows to objects.

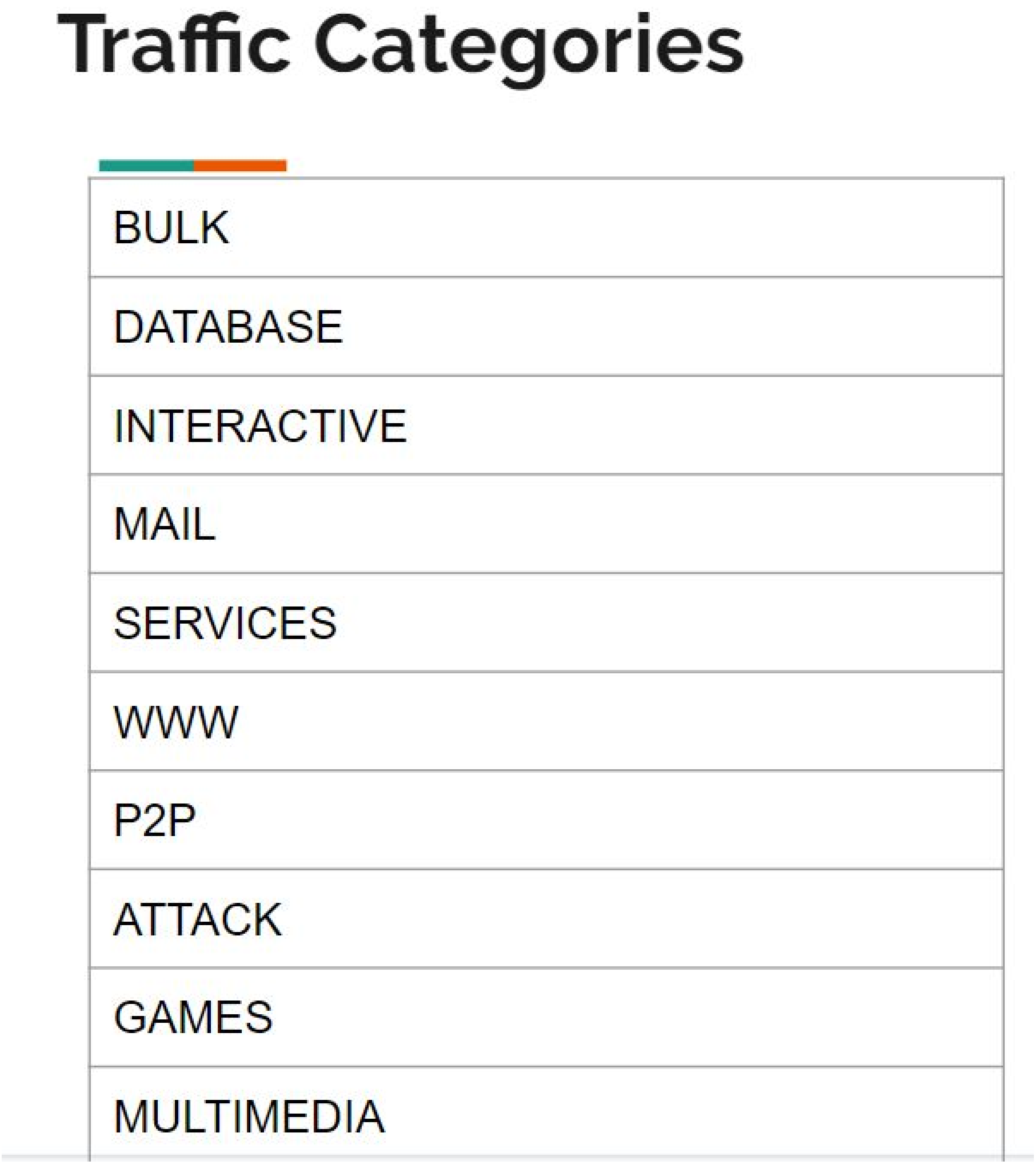


Discriminators Used

Flow duration
TCP port
Packet inter arrival time
Payload size
Effective Bandwidth based upon Entropy
Fourier Transform of packet arrival time

3.3 Traffic categories

Fundamental to classification work is the idea of classes of traffic. Throughout this process we use classes of traffic defined as common groups of applications. Other users of classification may have both simpler definitions, e.g., Normal versus Malicious, or more complex definitions, e.g., the identification of specific applications or specific TCP implementations . Table below lists the categories we use alongside a number of example applications. The application list given in this table is not definitive The use of such categories is also illustrative, allowing ready comparison with simpler port-only classification techniques. Importantly, while each flow is mapped to only one category, the characteristics of the traffic within each category are not necessarily unique. For example, the BULK category which is made up of ftp traffic consists of both the ftp control channel which transfers data in both directions, and the ftp data channel which consists of a simplex flow of data for each object transferred. The grouping of applications into the categories we have given is largely an artificial, user-centric grouping and further serves to illustrate that such arbitrary clustering of only minimally-related traffic-types is possible within our scheme.

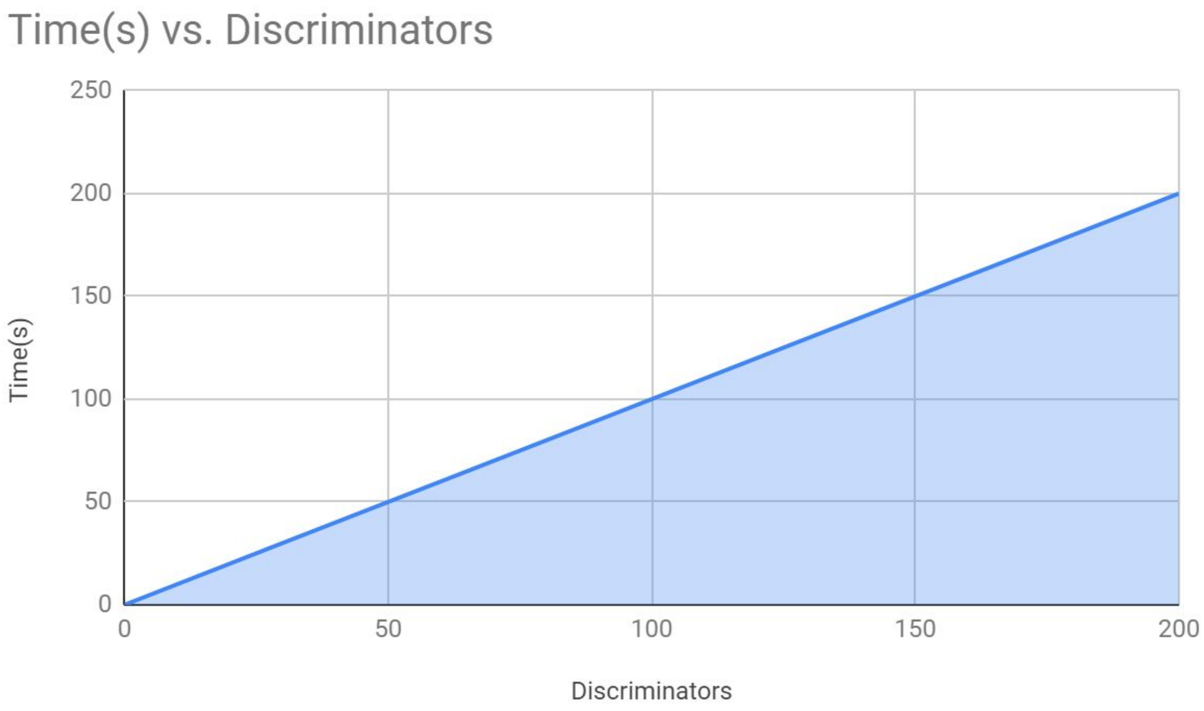


Traffic Categories

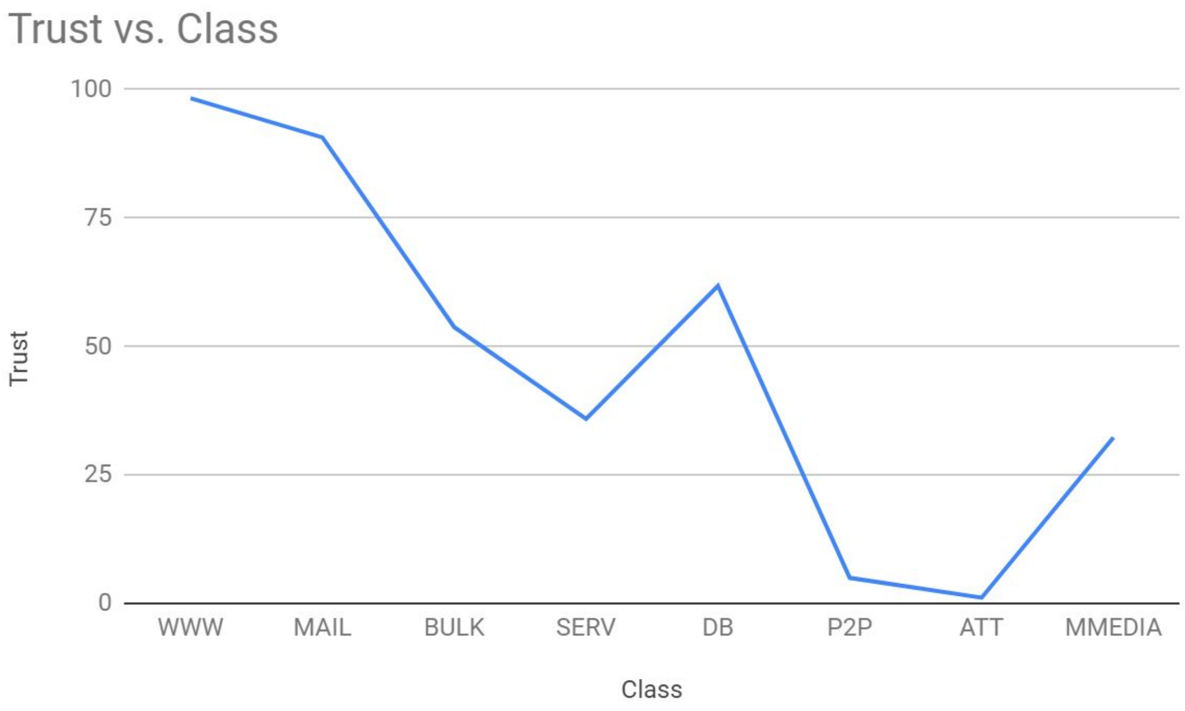
BULK
DATABASE
INTERACTIVE
MAIL
SERVICES
WWW
P2P
ATTACK
GAMES
MULTIMEDIA

4 Experimental result

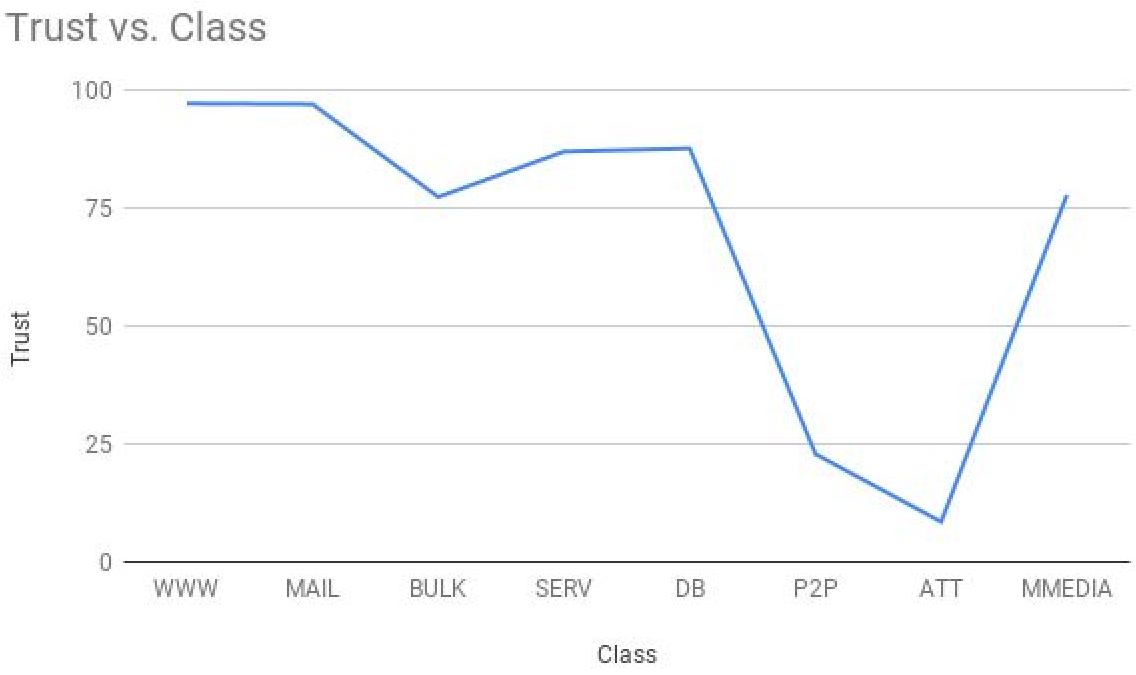
4.1 Time took to build a model on experimental training and testing set against discriminators



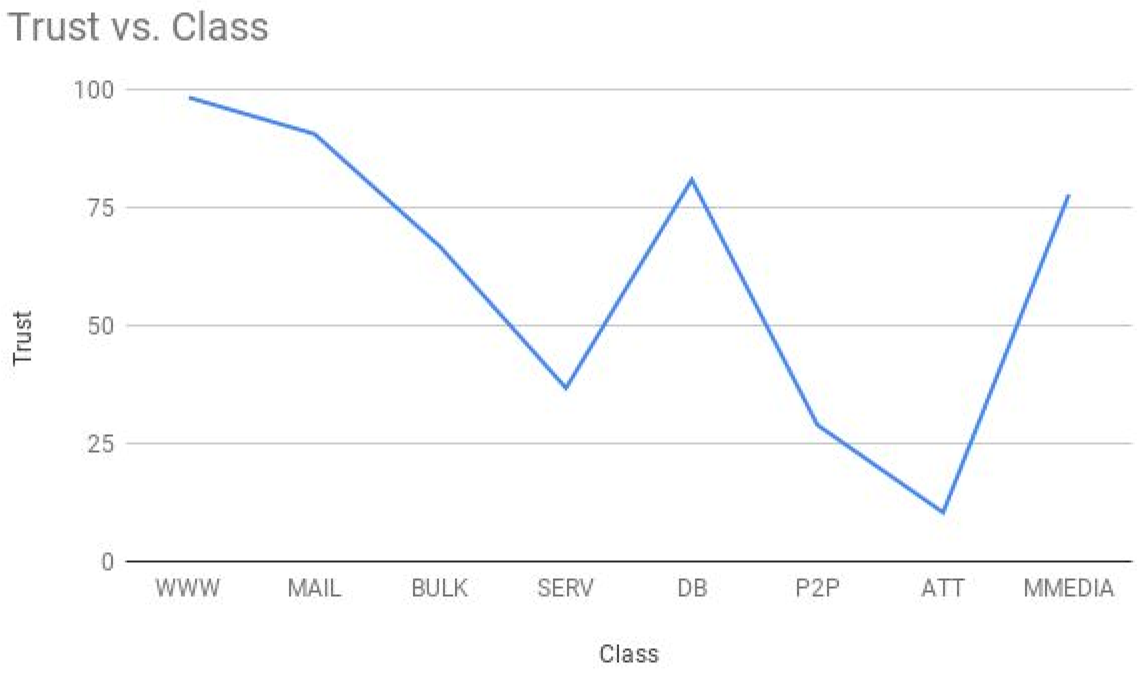
4.2 Naive Bayes approach to traffic described by all discriminators



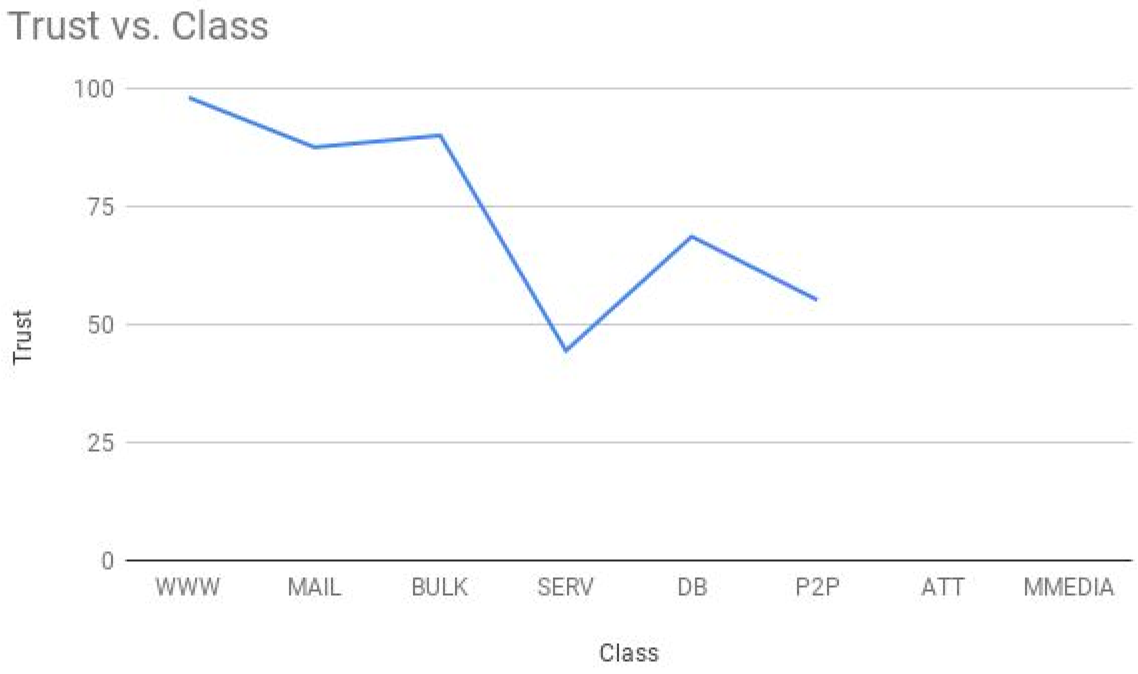
4.3 Naive Bayes method, with kernel density estimation, performed on all discriminators


Trust vs. Class
100
75
50
25
0
Trust
WWW
MAIL
BULK
SERV
DB
P2P
ATT
MMEDIA
Class

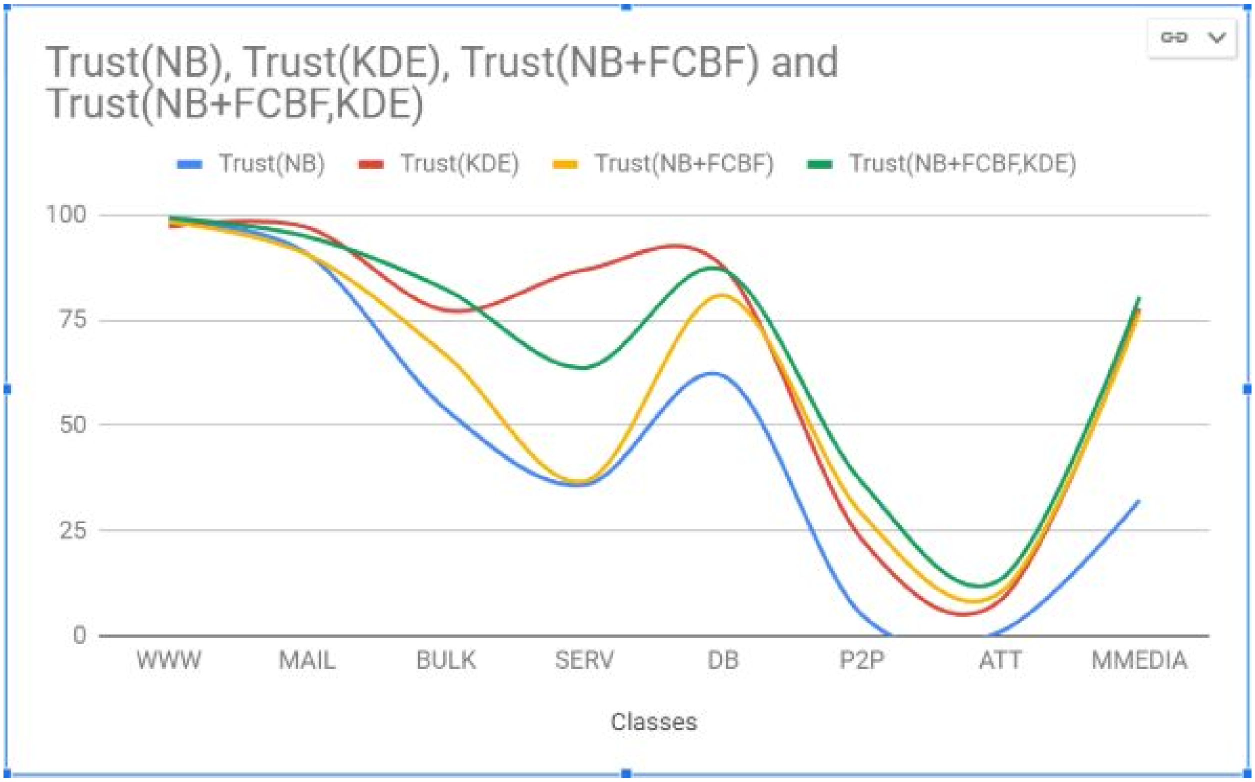
4.4 Naive Bayes, after FCBF prefiltering



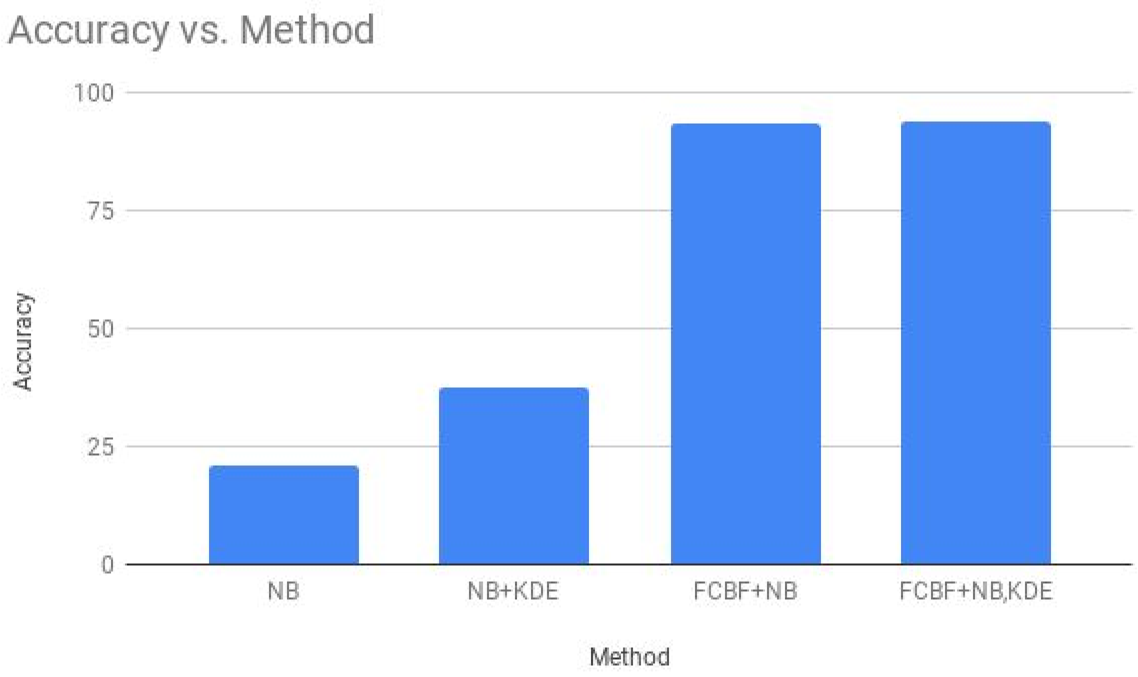
4.5 Naive Bayes, kernel density estimation technique after FCBF prefiltering



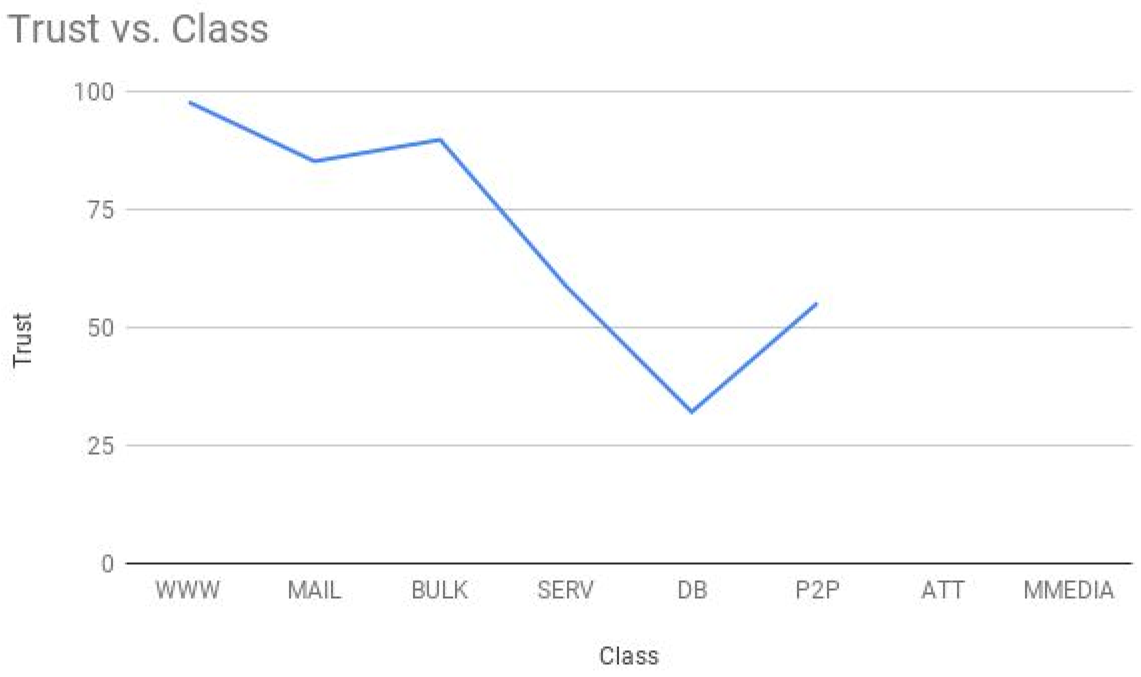
4.6 Comparison of all methods



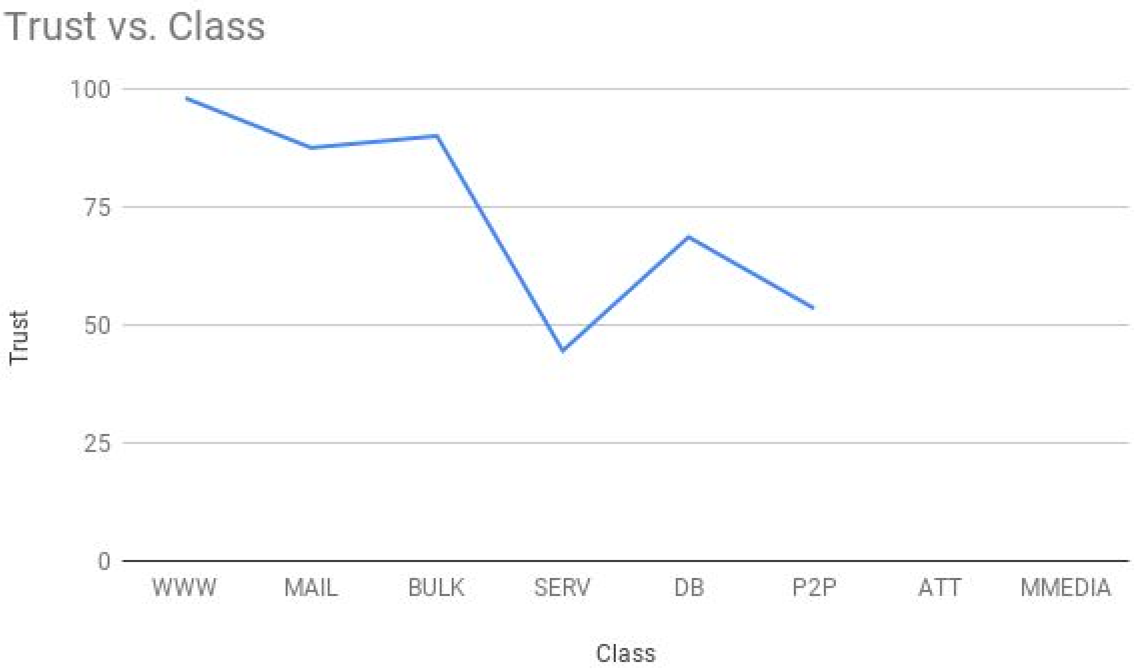
4.7 Average percentage of accurately classified flows by different methods. Evaluated with a dataset from a later time.



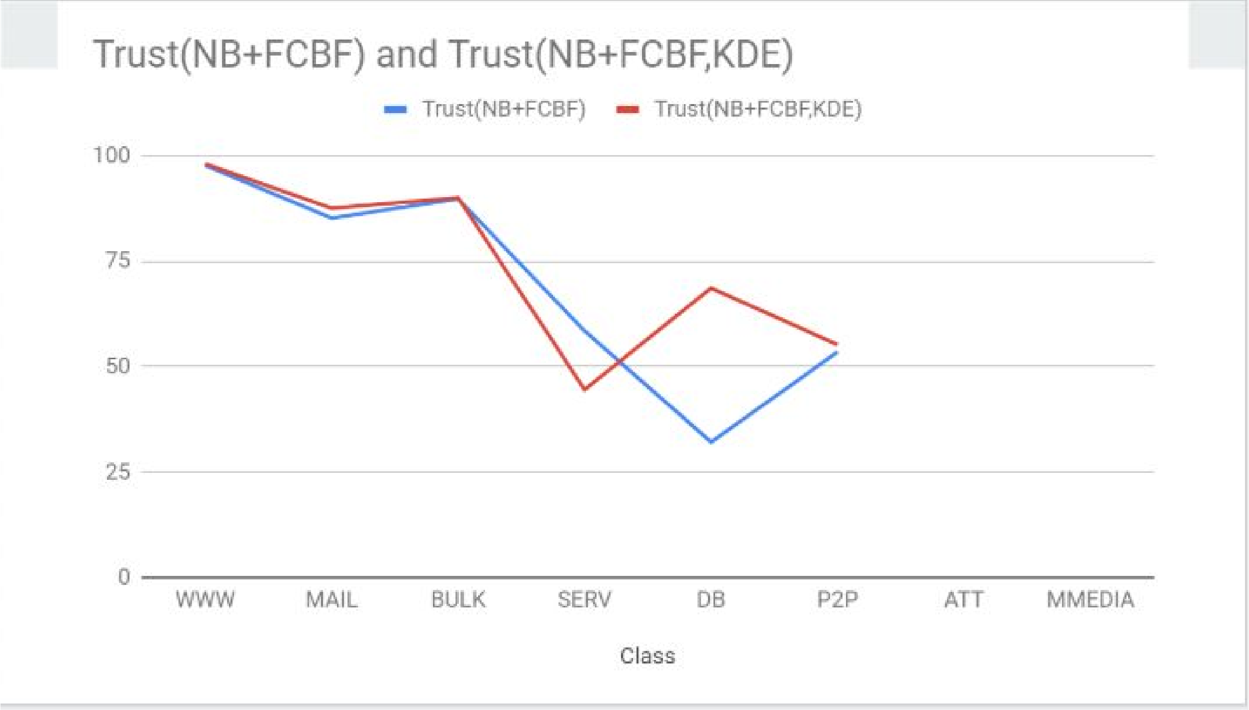
4.8 Measure of belief if a certain class occurred in the Naive Bayes method after FCBF prefiltering. Evaluated with a dataset from a later time.



4.9 Measure of belief if a certain class occurred in the Naive Bayes, kernel density estimation method after FCBF prefiltering. Evaluated with a dataset from a different time.



4.10 Measurement of belief in certain class after NB+FCBC,(NB+FCBC,KDE)(Comparison)



Last page of the Document

{Intentionally Left Blank}