Masquerade Detection: A Topic Model Based Approach

by

Jennifer Parkhouse

B.Sc., Simon Fraser University, 2013

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Fall 2018

Approval

Name:	**Jennifer Parkhouse**
Degree:	**Master of Science (Statistics)**
Title:	***Masquerade Detection: A Topic Model Based Approach***
Examining Committee:	**Chair:** Jinko Graham Professor

Derek Bingham
Senior Supervisor
Professor _____

David Campbell
Supervisor
Associate Professor _____

Tom Loughin
Examiner
Professor _____

Date Defended: December 19, 2018

Abstract

The goal of masquerade detection is to "detect" when an intruder has infiltrated a computer system by looking for evidence of malicious behaviour. In this project, I use a topic model based intrusion detection system to search for intruders within the SEA and Greenberg datasets of Unix computer commands. Using LDA topic modeling I was able to find a probability distribution for each user for both the topics over a block of commands and over each command. Using these two probability distributions and building on previous detection techniques I was able to create five different detection techniques. I describe how I created the five LDA based models and combine them to find a sixth model. All of these techniques performed on par with their non-LDA counter-parts. Therefore, combined with the reduction in dimensionality afforded by the LDA topic model, I conclude that my methods perform better than the current models.

Keywords: Intrusion Detection; masquerader; masquerade detection; latent direchlet allocation; topic modeling

Acknowledgements

I have faced many challenges while working on my masters and would like to thank everyone who helped me achieve this amazing feat of completing my masters.

In particular, I would like to thank Dr. Derek Bingham who took an initial interest in me in my undergraduate degree and continued to believe in me throughout my masters. He saw my potential and took a chance on me offering to take me on as a masters student after working with me on a undergraduate research project. I would like to thank him for having the patience and understanding when I needed to take time off and for inspiring me to study statistics in the first place.

Many thanks to Dr. David Campbell for his support and knowledgeable help on my masters project. I would also like to thank Dr. Gary Parker for fostering my interest in actuarial science, in particular, in my interest in completing the Graduate Diploma in Financial Engineering alongside my masters. I would finally like to thank the entire Statistics and Actuarial Science department including all the students and staff I met along the way. Without all of their guidance and support I wouldn't be where I am or who I am today.

Perhaps most importantly of all, I would like to thank my family for their undying support and love during this trying time in my life. I would not have been able to accomplish even half of what I have without you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With an increase in the number of companies moving data to computer systems and an increase in the quantity of information a company stores, the threat of cyber attacks has increased in recent years. For example, in 2015 the networking technology firm Ubiquiti Networks Inc. suffered a $46.7 million loss due to a form of cyber attack known as spear phishing and only managed to recoup a small portion of this loss (approx. $8.1 million) [12]. Any company which uses computer systems to access and store critical information is vulnerable to attack. Thus, security of computer systems and networks has become a main priority for many companies. Specifically, companies want to intercept or block intrusions into their computer system. Such intrusions come in many forms, for example, spoofing (impersonating other users), viruses, eavesdropping (interception of network traffic), or tampering of data. Most of these intrusions leave a trace in the log file which links the attack to a specific user. However, arguably, one of the more serious security threats to companies is a type of spoofing known as masquerading which doesn't leave a log file trace [17]. In particular, masquerading is carried out by a person or entity known as a masquerader who impersonates a legitimate user, typically by stealing the legitimate user's password, forging the legitimate user's email, or violating the system authentication, in an attempt to gain access to a computer system and carry out malicious behaviour. This malicious behaviour can range from disrupting operations or corrupting data to stealing sensitive information. In addition, masqueraders can either be outsiders, who gain access to the computer network via a legitimate user's identity, or insiders, who are legitimate users, but purposefully perform tasks which are malicious to the computer network. In practice, masqueraders are typically insiders as outsiders quickly try to gain access to the account of a super-user and are easily detected [22]. Therefore, detection of masqueraders is an area of great interest for today's companies.

The outline of my project is as follows. In Chapter 2, I provide some background information on intrusion detection, topic modeling, and Principal Component Analysis (PCA) and I discuss previous work done on intrusion detection. In Chapter 4, I introduce the topic

modeling methodologies I used to analyze the masquerader data, which is summarized in Chapter 3. In Chapter 5, I provide the results of the six methods. Finally, in Chapter 6, I discuss how the proposed method relates to previous methodologies and provide suggestions for future work.

The goal of this project is to design robust intrusion detection systems, which can detect masqueraders more efficiently and with less compute time than the current methodologies. This new intrusion detection system will work on host based systems and thus consider anomalies based on individual user command traces.

Chapter 2

Background and Literature Review

In this chapter, I briefly outline some of the most common masquerade detection techniques. I first give a brief overview of intrusion detection systems followed by a literature review of masquerade intrusion detection systems. Next, I discuss latent dirichlet allocation (LDA) topic modelling, the principal methodology underlying the intrusion detection systems proposed herein. In addition, although LDA has not, to my knowledge, been used for masquerade detection, I discuss some modern research in intrusion detection using LDA based methods. Finally, I provide some background on principal component analysis that is used in two of the proposed methodologies.

2.1 Intrusion Detection

2.1.1 Background

Ideally, I would like computer systems to be completely secure and block all forms of intrusions. However, most experts agree that complete security will never be reached [16]. Think of this as a hypothesis test that always makes the correct decision. Thus, the goal of intrusion detection is to detect when an intruder has entered a computer system. That is, "[i]ntrusion detection refers to the detection of malicious activity [8]". Strictly speaking, intrusion detection doesn't actually detect intrusions, but, rather, looks for evidence (as specified by the detection technique used) which may indicate that an intruder is in the system. The techniques and technologies used to perform intrusion detection are refered to as an intrusion detection systems (IDS).

Early intrusion detection systems focused on detection after the fact, performing intrusion detection at the end of the day when system activity was low. More recent intrusion detection systems focus on detection techniques which can be used immediately to stop an intrusion as it happens. In this project, I consider IDS which work immediately.

In addition, IDS fall into two categories: host based and network based. In host based, the IDS considers only one computer, or host, and detects intrusions based on operating

system call traces [8]. In contrast, network based IDS considers intrusions to a computer or computers via a network. Intrusion detection is done by considering network data such as packet traces. In this project, I are concerned with host based IDS.

Finally, there are two types of intrusion detection techniques: anomaly detection and misuse detection (also known as sequence matching). First, anomaly detection assumes that behaviour during intrusions differs from normal activity. Thus, anomaly detection looks for changes in a user's, or application's, normal behaviour. Second, misuse detection considers the behaviour of a pre-defined set of attacks and compares the attack behaviour to the current behaviour being tested. In comparison, anomaly detection allows for previously unknown attacks to be detected, while misuse detection only considers a set of known attacks. Allowing for unknown attacks leads to higher false positive (FP) rates as legitimate changes in behaviour will also be labelled as masqueraders [16, 24]. For this project, only IDS based anomaly detection is investigated.

2.1.2 Masquerade Detection Literature Review

DuMouchel [11] proposed a Bayes, one-step Markov model for masquerade detection. This method used a Bayes factor to test whether the one-step transition probabilities between test commands were consistent with a historic transition matrix from the training data. The method was found to have satisfactory behaviour when there are no masqueraders, but has low statistical power. That is, the false positive (FP) was close to the desired value, but the false negative (FN) rate was relatively high.

Later, Schonlau and Theus [21, 22] proposed a uniqueness method which considered detecting masqueraders based on the use of unpopular/unique commands. The authors posit that commands which have not been used before may indicate a masquerader and that the probability that a command is from a masquerader is inversely related to the number of users who use the command. Like DuMouchel's approach this method (referred to hereafter as the "uniqueness method") has a relatively low false positive detection rate, but also a high false negative rate.

Next, Wan et al [24] built on the uniqueness method by using high frequency commands rather than unique commands to detect masqueraders. In this method, two vectors are built, one with profile/training frequencies and one with signature/testing frequencies of the top n commands. These two vectors are then compared either directly or with smoothing. This method had a similar false negative rate to the Bayes one-step Markov, but had a very large false positive rate.

An adaptive Naive Bayes method was proposed by Dash et al [10]. This method considers deviations from normal behaviour to be suspicious if they are only temporary and masqueraders if the deviation continues for longer periods. In this way, blocks of commands are labelled as legitimate, doubtful, and masquerader. If a block is labelled as doubtful then

the following block has to go through a more rigorous test, and only if the previous 2-3 blocks have been classified as doubtful, will a block be classified as a masquerader.

Additionally, Camina et al. [5] proposed an intrusion detection system for masquerade detection based on file system structure and use. That is, the authors consider "how and what a user browses while working on her own file system [5]" rather than the actual commands. Later, Rodriquez et al. [19] built on the file system navigation model of Camina et al. and the use of tasks (a collection of interrelated files). The authors refined the notion of a task which was first introduced by Camina et al. in 2014 [6]. In particular, they created an additional single task which encompasses all elements which do not belong to the user specified tasks. This allowed for the large number of small tasks to be relocated to a single task and thus better classify the user [19].

Finally, a data-driven semi-global alignment (DDSGA) technique was proposed by Kholidy et al [17]. In this paper, the authors propose an improvement in both the computational and security efficiency of the Enhanced-SGA. In particular, the SGA algorithm matches large sequences of the test data to the training signature while still preserving local alignments. Then, Enhanced-SGA allows for changes to the training signature over time to account for changes in the user's behaviour due to things like changing roles. The improvement of DDSGA is to label areas of misalignment as anomalous and then signal an attack if the percent of anomalous areas is larger than a set threshold. In addition, DDSGA allows for user specific scoring parameters which increases the detection accuracy.

The performance of these methods can be compared on the SEA dataset (more on this later) as shown in Table 2.1. First, as mentioned previously, the uniqueness method achieves the lowest false positive rate at just over 1%, but the false negative is the largest. Next, the high frequency method betters the false negative rate by half, but the false positive is greatly increased. The Bayes, one-step markov method achieves a false negative rate similar to the high frequency method, and the false positive rate is halved. Next, the adaptive naive Bayes method achieves a false positive similar to the Bayes, one-step markov method, and the detection rate is greatly increased. Finally, the DDSGA method achieves a very low false positive rate, 3.4%, nearly half that of the adaptive naive Bayes method, while still maintaining a low false negative rate.

Technique	Detection Rate	False Positive Rate	False Negative Rate
DDSGA	83.3	3.4	16.7
Adaptive Naive Bayes	87.8	7.7	12.2
High Frequency	69.7	13.9	30.3
Uniqueness	39.4	1.4	60.6
Bayes One-Step Markov	69.3	6.7	30.7

Table 2.1: Comparison of previous masquerade detection techniques using the SEA dataset.

2.2 Topic Modelling - Latent Dirichlet Allocation (LDA)

2.2.1 Background

Topic models, are used to discover the main themes or topics of a large collection of unstructured documents, and the primary methodology underlying my proposed intrusion detection systems. LDA is the most commoly used approach for topic modeling. In order to understand LDA, it is important to first define some key terms. In particular, a *word* is a "basic unit of discrete data [3]", a *document* is a list or sequence of words, a *corpus* is a collection of documents, and a *topic* is a distribution over a fixed vocabulary of all words within a corpus. In the context of host based masquerade detection, a word is a unix command, a document is a group of commands from the same user and a corpus is a collection of command groups from a variety of users.

LDA is a probabilistic model for a corpus which assumes that documents are made up of a small number of latent topics [3]. In order to model the latent topics, LDA considers finding the hidden structure (topics, topic distribution per-document, and topic assignments per-document per-word) which generated the observed corpus. In particular, this can be thought of as reversing the assumed generative process (see Figure 2.1). That is, for a random document d, made up of words $w = (w_1, \ldots, w_N)$, it is assumed to be generated by the following steps. First, randomly choosing a distribution over topics (θ_d), in practice θ_d is the Dirichlet distribution. Second, randomly choose the proportions of the topic distribution ($\phi_{Z_{d,n}}$), the Dirichlet distribution in practice. Finally, for the n^{th} word within the document, randomly choose a topic ($Z_{d,n}$), in practice the topics follow a multinomial distribution, based on the chosen topic distribution, θ_d and given $\phi_{Z_{d,n}}$. It is important to note that LDA assumes a *bag-of-words* approach to document generation, in that each word within a document is generated independently and thus the order of the words doesn't matter. Formally, the LDA generative process corresponds to the following joint distribution,

$$p(\phi, \theta, Z, W) = \prod_{t=1}^{T} p(\phi_t) \prod_{d=1}^{D} p(\theta_d) [\prod_{n=1}^{N} p(Z_{d,n}|\theta_d) p(W_{d,n}|\phi_t, Z_{d,n})].$$

In particular, LDA assumes the following distributions,

$$\begin{aligned} \theta_d &\sim Dirichlet(\alpha), \\ \phi_t &\sim Dirichlet(\beta), \\ Z_{d,n} &\sim Multinomial(\theta_d), \text{ and} \\ W_{d,n} &\sim Multinomial(\phi_{Z_{d,n}}), \end{aligned} \tag{2.1}$$

where α and β are hyperparameters defined at the corpus level.

Computing the conditional distribution of the topics based on the observed documents (the posterior) is equivalent to doing the reverse of the generative process. Formally, the

posterior is denoted by

$$p(\phi, \theta, Z|W) = \frac{p(\phi, \theta, Z, W)}{p(W)}.$$

However, $p(W)$ is the sum of the joint distribution of every possible topic structure which is intractable [2].

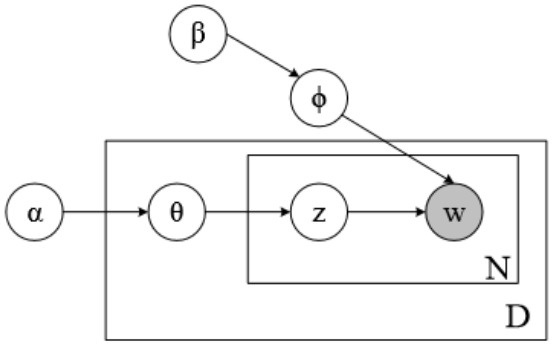


Figure 2.1: Graphical representation of the LDA assumed generative process for documents. ([7], Chapter 3, p. 3)

2.2.2 LDA Based Intrusion Detection Literature Review

Research pertaining to the use of LDA methods for masquerade detection has focused on broad network based intrusion detection techniques. In particular, initial research into using topic modeling to do intrusion detection was in [9]. The authors apply LDA and dynamic LDA (dLDA), a time evolving LDA, as a means of identifying underlying topics within a network based computer system. Although, they do not actually apply intrusion detection, they posit that topic modeling can be used as a means of reducing the dimensionality of the intrusion detection problem. Finally, they outline two possible ways to carry out intrusion detection after topic modeling has been done. First, they consider topic modeling both normal and malicious activity and then relate a subset of the topics to malicious activities. The authors note that topics wouldn't necessarily correspond directly to malicious activities and thus suggest that a sparse Bayesian classifier could be used. Second, they suggest using only normal activities to do topic modeling and then consider how well the topics explain new activities. This second approach is considered by the authors to be a more realistic tactic to intrusion detection using topics.

Similarly, in 2014, Huang et al [14] proposed a LDA based misuse detection technique. They suggest using event types (ie. unknown IP address, SSH connection) to create topics using LDA. These topics are created for both known malicious events and normal events as determined via clean training data. Once this is done, new logged events are assigned

probabilities of belonging to each topic which correlates to probabilites of being malicious or not. This approach, however, doesn't allow for new malicious events.

2.3 Principal Component Analysis (PCA)

2.3.1 Background

Principal component analysis (PCA), one of the most widely used multivariate techniques, has a variety of goals such as dimension reduction or searching for structure to simplify interpretation [1].

Extracting only the most important information in a dataset allows PCA to reduce the dimensionality of a dataset, which may contain a large number of correlated variables, while at the same time accounting for as much variability in the data as possible [15]. For dimension reduction, the original multivariate data is transformed onto a new smaller set of orthogonal variables known as *principal components* [1]. That is, a principal component is a linear combination of the original data. Note, the reduction in dimensionality of the data allows for easier interpretation when the number of kept dimensions is small, 2 or 3, as, graphically, PCA fits the original dataset into an 2 or 3 dimensional ellipsoid with axes corresponding to principal components.

Additionally, the principal components are sorted so that the first principal component accounts for the largest variability in the original dataset, and each successive principal component accounts for the largest variability given that the principal component is orthogonal to all the proceeding principal components [1]. It is important to note that when doing PCA, only a subset of the principal components are used, namely those that explain the most variability in the system. The number of principal components that are kept and used to describe the dataset depends on the desired amount of variability accounted for at any given time. If thought of in terms of the n-dimensional ellipsoid, these largest variability principal components, which are kept when doing the PCA, correspond to the longest axes. Therefore, it can be easily seen visually that not much information/variability is lost if the smaller axes or principal components are not used to describe the original dataset as they do not change the data very much.

Typically, prior to performing PCA the original dataset is standardized. This standardization is done to account for variables with differing scales and to allow each variable to receive equal weight in the subsequent analysis [15]. This amounts to conducting PCA on the correlation matrix of the data rather than the covariance matrix.

2.3.2 PCA Based Intrusion Detection Literature Review

Initially, PCA was used in intrusion detection only as a tool to reduce the dimensionality of the problem (See [4] for example). Later, Shyu et. al. [23] proposed a method which used

PCA as an outlier or anomaly detection tool on network data. They were able to distinguish deviations from normal behaviour which they identified as intrusions. The authors consider detecting anomalies using 2 methods. First, they detect large values compared to original features by using the principal components which explain around 50 percent of the total variation. Second, observations which have differing correlation structures are detected using the remaining components.

Later, Wang, Guan, and Zhang [25] proposed using principal component analysis as a means of detecting anomalies within a computer system. Specifically, the authors suggest using frequencies of system calls in a trace or of individual commands as input. Then, PCA is used to reduce the dimensionality of the problem. Finally, distances between the projection onto the principal components and the original data is used as a means of detecting anomalous behaviour. That is, normal behaviour assumes that the data and its projection are similar and thus the distance between them would be small. They consider three different distance measures; squared Euclidean, cosine, and signal-to-noise ratio (SNR). See Chapter 2 from [25] for a more detailed explanation of the distance measures. Using the SEA dataset, they observe a 100% detection rate. However, the results are based on only one test which used a combination of two users data out of the 50 possible so the detection rate is not accurate.

Further research on masquerade detection using PCA has taken a misuse detection approach and used methods similar to that of Wang [25]. Finally, most new research on intrusion detection is focused on network data and will not be discussed here as my project considers host based rather than network based systems.

Chapter 3

Data

For the purposes of the methodologies explored this project, there are two publicly available datasets: SEA and Greenberg. Both datasets contain blocks of Unix commands.

The SEA dataset is available from Schonlau's website [20]. The SEA dataset consists of audit data for 50 users. For confidentiallity, the audit data has been stripped of all identifying characteristics such that only the plain unix commands without arguments remain. Each user has 15,000 commands (some legitimate and some masquerader) organized into blocks of 100 commands in the sequence in which they occurred. These blocks are then broken into training and testing data. The first 5,000 commands (50 blocks) are training data and are free of any masqueraders. The remaining 10,000 commands (100 blocks) are testing data made up of both audit data from the user and randomly interspersed audit data (command blocks) from users outside the 50 included in the dataset. These randomly interspersed command blocks are meant to act as masqueraders. See Figure 3.1 for a graphical representation of the data.

Along with the SEA data is a file containing the location of the masquerade blocks. It is a matrix of 100-by-50 corresponding to the 100 testing blocks for all 50 users. The matrix is made up of 0's and 1's where 0 means that the block of data does belong to a masquerader.

User i		Command Number					
		1	2	3	...	99	100
Training Blocks	1	cpp	kill	xhost			
	2	sh	mail				
	..						
	50						
Testing Blocks	51						
	...						
	149						
	150						

Figure 3.1: Graphical representation of the SEA dataset for random user i with some toy data filled in.

The Greenberg dataset [13] contains unix commands from 168 users, broken into four categories of users (novice programmers, experienced programmers, computer scientists, and non-programmers). The data also contains information pertaining to session start and end times, unix command aliases, history use, current working directory, and any errors.

Since the Greenberg dataset contains extra information and no masqueraders some pre-processing of the data was required. First, I are not interested in the different user categories so all users were grouped together. Second, I are only interested in the raw unix commands so all extra information pertaining to alias etc was disregarded. Third, I introduced masqueraders into the data by using a portion of the users as masqueraders and inserting some of their command blocks into the other users' testing blocks. In particular, I followed the method of Dash et al [10]. That is, users were kept to be used for testing the methodologies if they had between 2000 and 5000 commands, otherwise their blocks of commands were used as masquerader blocks. Next, the users commands were truncated to 2000 and the first 800 were used as training data. The rest of the commands were treated as test data with 10 commands per block. Then, ten masquerade blocks were randomly chosen and inserted into the test data where the masquerade blocks were of size 30, but once inserted the blocks were treated as size 10. In this way, there were always three consecutive masquerade blocks of size 10, which is needed for one of the methods.

Chapter 4

Methodology

In this project, new methodologies are proposed and evaluated to see if I can improve the current IDS methodologies. In this chapter, the LDA topic model is described, innovations to intrusion detection methods are proposed to describe how I created and used a LDA topic model, and how I combined these five methods to try to improve the results of any one individual method is discussed. Finally, the approach for validating the thresholds from all five methods is described.

4.1 Topic Model

First, I performed topic modeling by considering blocks of training commands from all users to create one topic model for all users. In particular, I treated each block of commands as a separate 'document' and the individual commands as 'words'. Then, I used a variational expectation-maximization (VEM) topic modeling procedure with a given number of topics, k.

Using the trained topic model, a probability distribution over the topics for each test block was attained. Therefore, each user had the number topic distributions equal to the number of training blocks. For example, using the SEA dataset, all 50 training blocks from all 50 users are combined to make one corpus of 2500 blocks and a topic model is run on this corpus of 2500 blocks. Then, for each of the user's 100 test blocks a probability distribution is obtained such that the probability of topic i occuring in block b from user u is $p_{i,b,u}$. The first method, PCA on topic probability, uses these probabilities as the data and then runs a PCA on them (more on this later).

Similarly, the topic model gives a probability distribution over the topics for each command. Then, I assigned each command to a single topic. In this way I reduced the dimensionality of the problem from a large number of commands to a small number of topics. Furthermore, similar commands or commands which are more likely to occur together were clustered into one topic prior to performing masquerade detection.

First I separated the commands into three categories; (i) used in the topic model, (ii) used in the training data, but too rare to include in the topic model, and (iii) only used in the test data. Next, I assigned all commands in categories (ii) and (iii) to separate topics which increased the number of topics from k to $k + 2$. Then, for commands used in the topic model I considered doing this assignment using three algorithms (see Algorithm 4). The first algorithm, best topic, assigns the command to the topic where the command has the highest probability of occuring. The second algorithm, topic weights, randomly assigns a command to a topic using weights which are proportional to the probabilities associated with the command occuring in each possible topic. The third algorithm, coin flip, randomly assigns a command to a topic by 'flipping a coin'. That is, each possible topic, where possible means that the probability of assigning the topic to the command is above some threshold, has the same likelihood of having the command assigned to it.

4.2 PCA

Moving onto specific methods of anomaly detection considering two different modifications to the Principal Component Analysis anomaly detection methodology proposed by Wang [25]. Before getting to the specifics of how I modified the methodology, the basic steps needed for the methodology of Wang are: (1) given a set of training data $x_1, \ldots, x_m$, calculate the mean vector μ of the training data and the set of mean-adjusted test, or validation, data, $(t_1 - \mu), \ldots, (t_n - \mu)$, where, t_j is the j^{th} test data point; (2) perform Principal Component Analysis on the training data to identify the set of most influential eigenvalue-eigenvector pairs $\{(\lambda_i, u_i)|i = 1, \ldots, k < m\}$; (3) these k eigenvectors are then used to form a n x k matrix U^T; (4) calculate the projection of the mean-adjusted test data onto the principal component subspace, U, denote this by y; (5) calulate the distance between the original test data and its projection, $\Phi - \Phi_f$, using

$$\begin{aligned} \Phi &= t_i - \mu, \\ y &= U^T \Phi, \\ \Phi_f &= Uy; \end{aligned}$$

and finally, (6) the authors then proposed that test data is from the user if the distance between it and its projection is small. Using the squared Euclidian distance, a test block was deemed to be a masquerader if ϵ was below a threshold, where

$$\epsilon = ||\Phi - \Phi_f||^2.$$

4.2.1 Method 1: Topic Probability

We built on the above methodology of Wang et al. [25] in two ways. First, I used my topic model rather than commands prior to performing PCA. Second, I performed PCA on both the test block topic distributions and the test block topic frequencies to create two different methods. Finally, each user had a different threshold, θ, at which the test data was considered a masquerader. To determine which threshold to use, I chose the threshold which resulted in the smallest distance between the false negative and false positive rates while still maintaining a false positive rate below 20%.

For this first method, by using topic distributions rather than command frequencies for the PCA (see Algorithm 1) I modified the methodology of Wang. Afterwards, I proceeded as in the original paper and used the eigenvalue-eigenvector pairs which accounted for 99.9% of the total variation and the squared Euclidean distance.

4.2.2 Method 2: Individual Command Topics

For this second modification to the methodology of Wang [25] outlined above, first, I used the same definition of 'frequency' as Wang used in his paper. That is, I count the number of times a topic occurs within a block and then divide by the length of the block (ie. 100). The frequency of the individual commands rather than the occurance of a command alone was used to define an occurance of a command. Thus, the frequency of the topics is the frequency of all associated commands. I calculated the training frequencies by aggregating over the 50 training command blocks.

Next, I proceeded as in Wang [25] and calculated the principal components of the topic frequencies. As before, I chose to use the eigenvalue-eigenvector pairs which accounted for 99.9% of the total variation and the squared Euclidean distance to be consistent with the original paper. For an outline of the method see Algorithm 1.

4.3 Frequency of Topics

The next two methods build on the work of Schonlau [22] and Dong Wan et al [24] by incorporating topic modeling into frequency based detection techniques. These two methods will be refered to as "high frequency" and "low frequency" methods. That is, I initially performed topic modeling as a way of grouping commands into specific topics and then I performed frequency based detection methods. Thus, I detected masqueraders based on anomalies within the frequencies of topics rather than frequencies of individual commands.

We use the topics associated with the individual commands attained from the topic probability distribution of commands. Using these $k+2$ topics, I calculated the frequency of topics within the training and testing data for each user. Specifically, I calculated the within block probability of a topic using the training data and compared this to the frequency of

topics within each test block. To do this, I first calculated the topic frequency within each training block using the frequencies of the associated commands. Secondly, I summed up all of the frequencies and divided by the number of blocks the topic occured in to get an average frequency per block. Third, I divided this average by the total number of blocks within the training dataset to get the probability of a topic occuring within a block.

In the high and low frequency methods I looked for dissimilarity between the training topic probabilities and each test block's topic frequency by performing a χ^2 test with a null hypothesis that there was no dissimilarity between the training probability and testing block frequency. If the χ^2 test did indicate evidence of a difference then the test block was marked as a masquerader.

4.3.1 Method 3: High Frequency

For this high frequency method, I only considered the top n_H topics which occured with high frequency when performing the χ^2 test. In addition, high frequency topics were considered to be any topic which occured with probability greater than zero within the training dataset. I allowed for each individual to have a different $n_H \leq$ the number of high frequency topics. I choose the 'optimal' n_H to be the number of topics which resulted in the smallest distance between the false negative and false positive rates while still maintaining a false positive rate below 20%. Therefore, I could achieve a good detection rate and false positive rate.

4.3.2 Method 4: Low Frequency

Similar to the previous high frequency method, I considered using the bottom n_L topics which occured with low frequency when performing the χ^2 test. For this method I needed to consider both n_L and a threshold for what is considered to be a low frequency topic, $f_L \geq 0$. Therefore, for this method, each user had two thresholds to optimize over. Otherwise, the method proceeded as with the high frequency method and similarly considered the 'optimal' (n_H, f_L) threshold pair to be the one which resulted in the smallest distance between the false negative and false positive rates while still maintaining a false positive rate below 20%.

4.4 Method 5: Combination of Methods

I tried to combine the above four methods in an attempt to improve the average rates. First, I combined the methods by looking for an agreement of the methods. That is, a block was considered to be from a masquerader if at least two of the methods indicated that it was a masquerader. In this way, all of the methods were considered to be equally good at predicting masqueraders.

4.5 Method 6: Adaptive Naive Bayes

The next method I modified was the Adaptive Naive Bayes method proposed by Dash et al [10]. This method allows for short term changes in behaviour which are legitimate, but marks long term changes in behaviour as a masquerader. This is achieved by marking a block of commands as legitimate, doubtful or masquerade. A block is marked as doubtful if it doesn't match with the training data and preceeding blocks were legitimate. If the preceeding two blocks were also doubtful then all three blocks are marked as masquerade. Also, since a masquerade block is more likely followed by another masquerade block, once a block is marked as doubtful the following block has to pass a more stringent test. Finally, if a doubtful block is followed by a legitimate block, then it is assumed that the doubtful block was a legitimate deviation in behaviour by the user and is changed to legitimate. I modified this methodology by using topic frequencies rather than command frequencies. For a complete overview of the method see Dash et al [10].

4.6 Cross Validation

All of the methods used in this project require at least one threshold. These thresholds have been chosen to obtain the best possible results given the SEA dataset or the Greenberg dataset depending on which dataset is being used at the time. In addition, I have used a subset of the dataset to demonstrate the usefulness of these methods in general. In particular, using the original topic model, I repeatedly randomly chose 45 of the possible 50 training blocks and used these to train each method. Next, for each randomly chosen training set, I randomly chose 100 test blocks from all users and calculated the best possible threshold. The test block selection was repeated a number of times such that for each training set there was a set of thresholds. Then, I used the given method with the random training set, the set of all possible thresholds, and the original testing set to obtain a matrix of masquerader indicators for each block (see Table 4.1). Next, I used a consensus method to determine if a block was a masquerader or not. That is, I considered a block of commands to be a masquerader if more than half of the test thresholds indicated that the block was a masquerader. This resulted in a vector of masquerader indicators for each training set. Thus, I used the same consensus method to calculate an overall masquerader vector for each user and then caluclated the false negative and false positive rates. See Algorithm 3 within the appendix for pseudocode of this validation method.

User i - Training Set j

Test Set	Block 1	...	Block 100
1	1	...	0
⋮			
k	0	...	1

Table 4.1: Example of matrix of masquerade indicators for a given user and training set for the cross validation method.

Chapter 5

Results

This section compares the six masquerade techniques and outlines the results of the six masquerade detection methods using both the SEA dataset and the Greenberg dataset. The optimal thresholds for the SEA dataset are shown in Table E.1. For each of the methods I consider a false positive (FP) to be when a test block is incorrectly identified as a masquerader, a false negative (FN) to be when a test block is incorrectly identified as a user, a true positive (TP) to be when a test block is correctly identified as a masquerader, and a true negative (TN) to be when a test block is correctly identified as a user.

To compare the methods in a real-world setting, I consider the average FP, FN, TP, and TN rates (hereafter refered to as 'average rates') over the 50 users in the SEA dataset and the 41 users in the Greenberg dataset. First, I compare the average rates for a method to the average rates of the same method using the actual commands rather than the topics. In this way, I can see if there is a benefit to using topic modeling beyond a reduction in dimensionality. Second, I compare the rates of my methods to the rates of the DDSGA method [17] as this is the method that achieved the best rates. Finally, I compare the rates from all of my methods against one another.

Lastly, I look at the computational times of each method. I compare the original IDS method to the novel topic based extension when running the SEA dataset with predefined thresholds.

5.1 Topic Model

For the SEA topic model, I have 50 users which have 856 words, 2500 documents and use 80 topics. This gives 100 topic distributions. For the Greenberg topic model I have 41 users with 2400 words, 3280 documents, and I use 80 topics.

5.2 Mahalanobis Distance

Since the two datasets that I are using use simulated masqueraders by inputting another users' data into one of the test users' data, it could be the case that the user and the masquerader would have similar sets of commands. In this case, the methods proposed here will not be able to detect the masqueraders as there is not enough of an anomaly within the data. Is it fair to say, then, that a method failed to detect the masquerder, or is it more appropriate to conclude that the masquerader isn't really a masquerader at all but simply another instance of a similar user? To determine whether any of these similar masquerade blocks are found within the datasets I need a measure of dissimilarity of the user and their masquerade blocks. FOr this, I used the Mahalanobis distance.

The Mahalanobis distance is the distance between two points in multivariate space. For uncorrelated variables, the Mahalanobis distance is the same as the Euclidean distance. However, for correlated variables, Euclidean distance no longer makes sense, but the Mahalanobis distance works even for correlated points. The Mahalanobis distance measures the distance of a point relative to the centroid of the multivariate data. Larger Mahalanobis distances mean the point is further from the centroid.

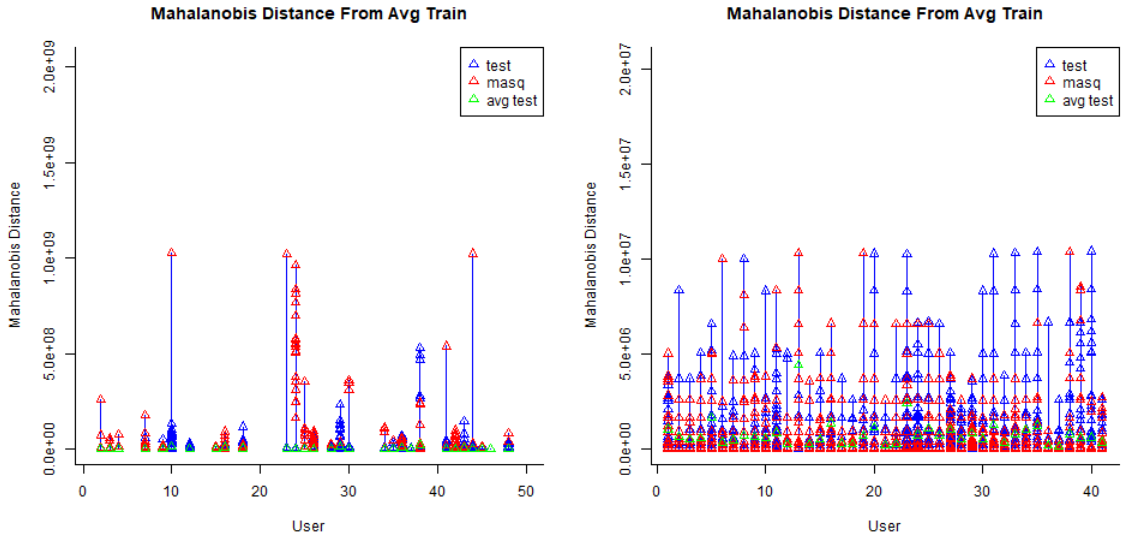
In the case of the SEA dataset, I treat the distribution to be the topic probability distribution from the training set of commands for each user. Then, I calculate the Mahalanobis distance from the mean of the distribution to the point corresponding to the count data for the masquerade blocks of count data. That is, for each user I treat only the 50 training blocks as the distribution made up of count data (either commands or topics) and calculate the Mahalanobis distance for each masquerade block within the users 100 test blocks. I want these masquerade blocks to have large distances indicating that they are truly dissimilar to the users expected behaviour, ie. an anomaly.

For the Greenberg dataset, since I are introducing the masqueraders ourselves I want to do so in a meaningful manner. Rather than randomly selecting a user and then randomly choosing a block of commands to insert I can calculate the Mahalanobis distance so as to choose a user and command block which is far from the expected users blocks. In particular, I first treat each users' count of number of commands which occur in a block as separate distributions (both training and testing blocks), then I calculate the distance between all users mean and all possible masqueraders mean count. Next, for each user and for each inserted command block acting as a masquerader, I randomly select a masquerade user with probability proportional to the Mahalanobis distance. Then, I select the masquerade blocks randomly from the chosen masquerader.

Figure 5.1 shows the Mahalanobis distance of each test block topic counts from the average of the training blocks topic counts for users with masqueraders. As seen in Figure 5.1a the Mahalanobis distance for the SEA dataset has masqueraders which are quite dissimilar compared to the rest of the test blocks. This indicates that there is a large enough

difference in the masquerader blocks to potentially be detected by the IDS. Not all users have these large distances though thus some users may be harder to detect masqeurader blocks than others. See Table F.1 for exact distances for each users' masquerade block from the training centroid and the mean of the test blocks Mahalanobis distance. For the Greenberg dataset, Figure 5.1b, on the other hand, the masquerader block distances to the non-masqueader's test blocks are relatively small. This may provide some evidence that my methods will struggle to detect these masqueraders (ie. high FN rate). There are several users who have test blocks that are a large distance away from the average training block which may be falsely identified as masqueraders. See Table F.2 for exact distances.

The Greenberg dataset points out the limitations of the methodology. When the masquerader looks like a user or when the test blocks don't look like the user, then it is difficult to correctly identify the masqueraders. Therefore, before using such approaches, it is important to perform similar analyses and define what sort of intruder one expects. Only then can one decide whether or not the methodology is likely to be effective in specific settings.



(a) SEA Dataset

(b) Greenberg Dataset

Figure 5.1: Mahalanobis distance of test blocks from average train block point.

5.3 Evaluation of Individual Methods

In the following sections, I compare the methods first on the SEA datset, followed by the Greenberg dataset. I begin with the PCA based methods, followed by the frequency based methods, then the combination method, and, finally, the adaptive naive Bayes method.

5.3.1 Method 1 and 2: PCA Based

SEA Dataset

The two PCA methods were both applied to the SEA dataset using the thresholds shown in Table E.1. The results are summarized in Table 5.1. Looking at the table, I see that there is little change in the average false positive rate when using command frequencies rather than topic frequencies. However, there is a small benefit provided for the false negative rate. On the other hand, when using topic probabilities I achieve a decrease in average false positive rate over the two frequency based methods, but see a large increase in average false negative rate.

Overall, there is evidence that using my PCA Topic Frequency based detection method has a minor improvement over the PCA Command Frequency based detection method presented by Wang [25] when considering the average rates. When combined with the fact that the proposed method also reduces the dimensionality of the problem and thus the compute time, I can see that my PCA Topic Frequency method may be preferable.

It is worth noting that I observe an improvement in the false negative rate, the false positive rate achieved by the proposed method is not as good as the DDSGA methods. The topic probability PCA method, on the other hand, does well if false positives are more severe than false negatives.

Next, Figure 5.2 shows the individual user's FN and FP rates. First, when I look at which method(s) gives the best false negative rate for each user (Figure 5.2b), I see that only three users have best FN rates which are achieved by a unique methods rather than having multiple methods giving the same FN rate. In particular, two users (users 7 and 36) achieve the lowest FN rate using the PCA topic frequency method and one user (user 3) has the PCA command frequency method resulting in the best FN rate. However, the FN rate from the PCA topic frequency method for user 3 is only slightly greater than the best FN rate (see Figure 5.2d). All other users have at least two methods resulting in the same FN rate. Furthermore, of these users, all of the best methods come from the PCA topic frequency method. The PCA topic probability method, on the other hand, gives a lot of FN rates well above the other two PCA methods and rarely results in the best FN rate.

Finally, it is important to note, two users (user 12 and 16) have a FN rate of nearly 100% for all three methods. This means that all three methods miss all of the masquerade blocks. Looking more closely at user 12 reveals that all but one of the masquerade blocks have around 20% of their commands coming from topic 46. However, when I look at which other blocks have a similar percent of commands from topic 46 I find that the percent of commands belonging to each block is similar for all of the blocks (including the masquerade ones). Therefore, the masquerade blocks are indistinguishable from the users true blocks. The same can be said for the one masquerade block with a low number of commands from topic 46. This can be seen in Figure 5.3a where all of the masquerader blocks are close to

the x-axis indicating that the masquerader blocks are all close to the training average block. Therefore, when doing PCA these masquerade blocks would look similar to the users blocks and thus won't be able to be found.

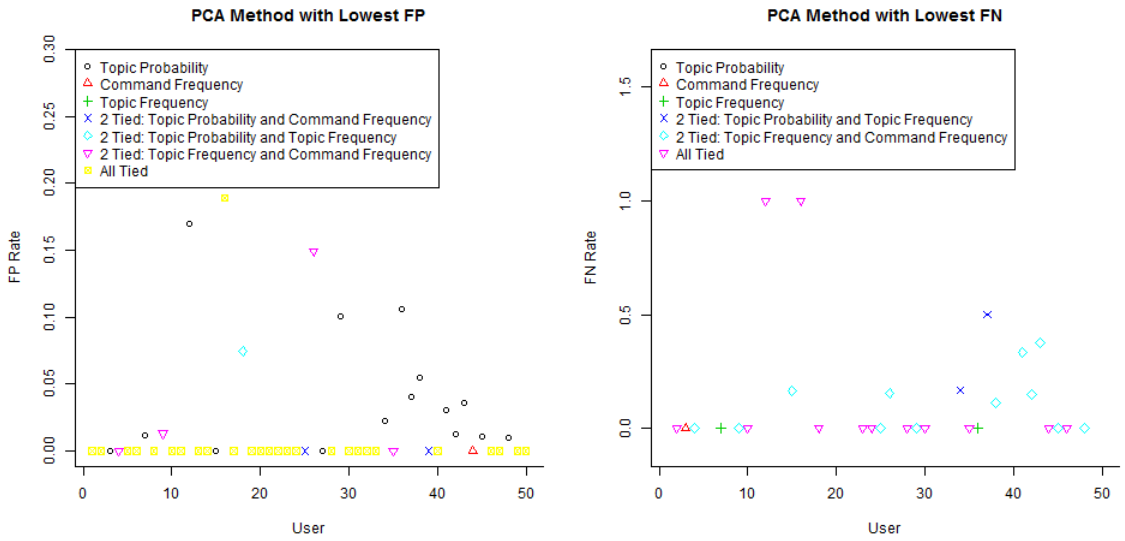
Simarly, user 16 has most of its masquerade blocks close to the average training block as seen in Figure 5.3a. However, there are four masquerade blocks which stand out as being far from the training blocks and from the other testing blocks. Therefore, it is surprising that these blocks are missed. Next, I look at the commands which occur (and thus the corresponding topics) compared to the other testing blocks and also the training blocks (see Figure 5.3b). From here I can see that the masquerader blocks are consistent with the training data and thus they could have easily been misslabeled as users rather than masqueraders.

Next, I look at Figure 5.2a which shows which method(s) result in the lowest FP rates for each user. Unlike the best FN rate, here I can see that half of the users have their best FP rate from the PCA topic probability method. Furthermore, only 8 users achieve their best FP rate from methods other than the PCA topic probability method. However, looking at the FP rate from the PCA topic probability method for these 8 users (see Figure 5.2c) I see that the best rate and the PCA topic probability rate are very close. Therefore, for all users the PCA topic probability method results in the best (or close to it) FP rate and thus does a overall better job than either of the other two methods. It is also important to note that all of the users have a good false positive rate of under 20%.

Finally, Table 5.1 shows the average rates obtained by the cross validation method outlined in Algorithm 3. Both topic based methods have low false positive rates when performing the cross validation which is consistent with the results I found using the entire SEA dataset. The false negative rate for the topic probability based method, however, is vastly greater for the cross validation.

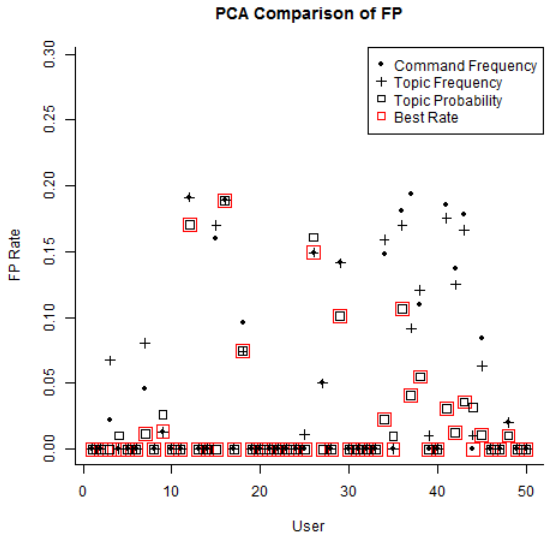
	TP	FP	TN	FN
PCA with Command Frequencies	82.93	4.60	95.41	17.07
PCA with Topic Frequencies	86.04	4.50	95.50	13.96
PCA with Topic Probabilities	61.54	2.22	97.78	38.46
PCA Topic Freq CV	86.07	2.22	97.78	13.93
PCA Topic Prob CV	25.02	2.22	97.78	74.98

Table 5.1: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the three Principal Component Analysis methods using the SEA dataset. Also included are the cross validation averaged rates.

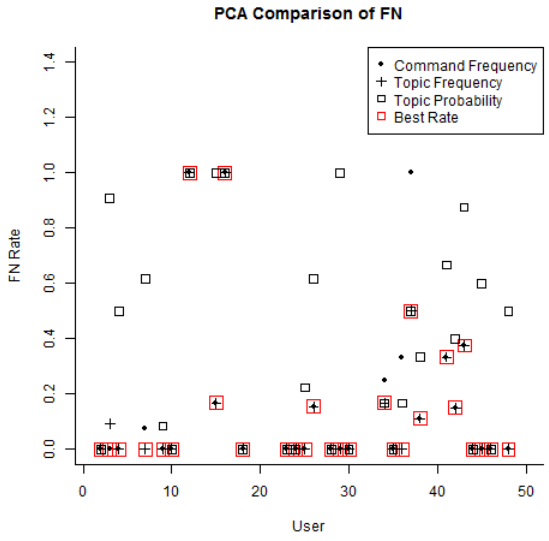


(a)

(b)

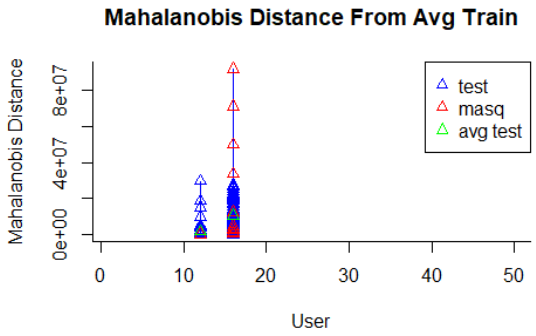


(c)

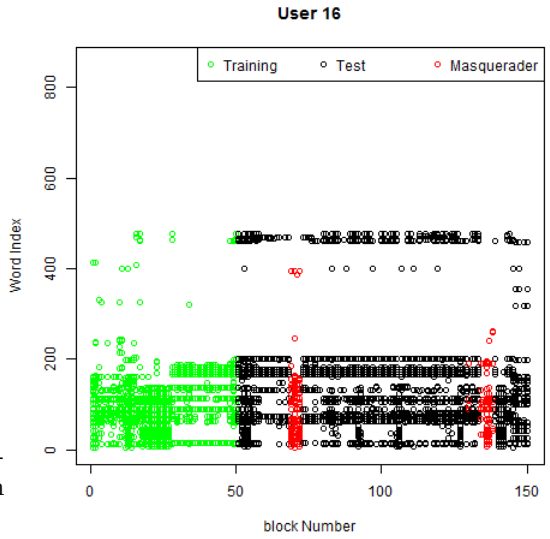


(d)

Figure 5.2: Comparison of false positive/negative rates for the three PCA based intrusion detection methods.



(a) Mahalanobis distance of test blocks from average train block point for User 12 and 16 which had 100% FN rates.



(b) Command occurance by block

Figure 5.3: Closer look at the users which have high FN rates for all three PCA methods.

Greenberg Dataset

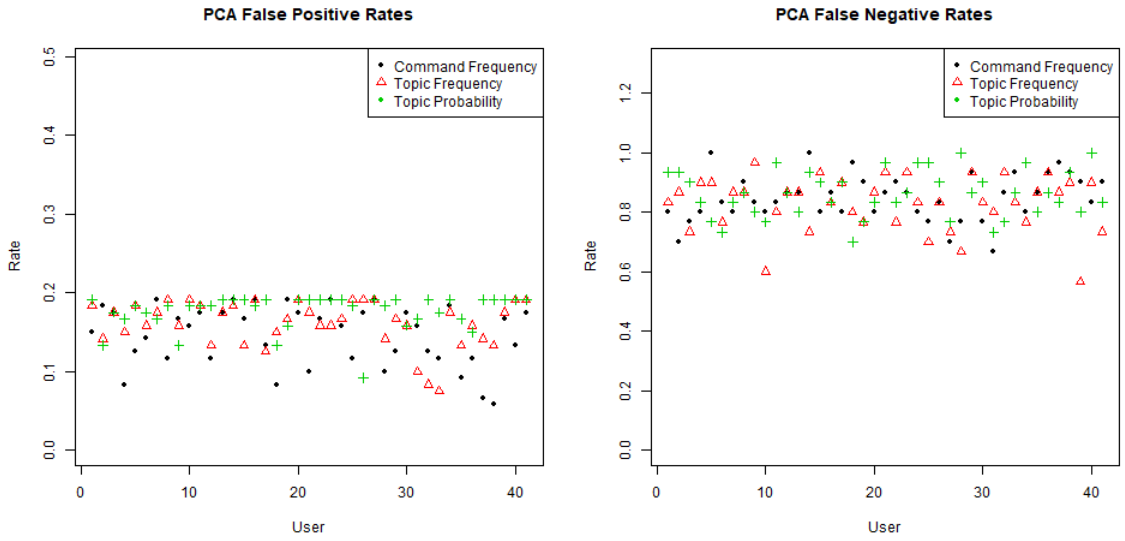
Now I repeat the analysis of the three PCA based methods using the Greenberg dataset. First, Table 5.2 shows that all three methods have similar average rates (around 16% FP and 85% FN). Therefore, none of the methods were very good at detecting masqueraders as the FN rates are very high. However, with the reduction in computational effort from the topic model I have evidence that my methods perform slightly better than the command based counter-part.

Second, I look more closely at which users had high or low FN and FP rates. To do this I look at Figure 5.4. I see from Figure 5.4a that all the methods have FP rates at or below 20% and the topic probability method has most users with FP rates just below 20%. Next, Figure 5.4b shows the FN rates for the three methods for all users. Here I see that all users with all methods have false negative rates between 60% and 100%. No users stand out as doing particularly well in terms of FN or FP rate no matter which method is used. This low power is as I expected as the Mahalanobis distances showed that the Greenberg dataset had test users which were not dissimilar from the masqueraders.

Finally, I look at the cross validation of the two topic based methods. See average rates in Table 5.2. Both topic based methods cross validations achieve a false positive of around 16-19% and a false negative of approximately 80-85%. This is as I expected from the initial results using the entire dataset.

	TP	FP	TN	FN
PCA with Command Frequencies	15.31	14.59	85.31	84.69
PCA with Topic Frequencies	17.26	16.10	83.90	82.74
PCA with Topic Probabilities	13.93	17.70	82.30	86.07
PCA Topic Freq CV	18.66	17.70	82.30	81.34
PCA Topic Prob CV	15.43	17.70	82.30	84.57

Table 5.2: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the three Principal Component Analysis methods using the Greenberg dataset. Also included are the cross validation averaged rates.



(a) (b)

Figure 5.4: Comparison of false negative and false positive rates for individual users based on the three PCA methods using the Greenberg dataset.

5.3.2 Method 3: High Frequency Topics

SEA Dataset

From Table 5.3 I can see that the original high frequency commands method does an overall better job then the high frequency topics method when considering the average users' false positive rates. The reduction in dimensionality gained from the high frequency topics method, and thus the reduction in compute time, may be enough to consider this method to be superior or as good as the command based method.

Next, when considering individual users, to see where the method does well and where it doesn't, I look at figure 5.5. From figure 5.5a I see that the high frequency topic method has a higher false positive rate for most users. It is also important to note that two users

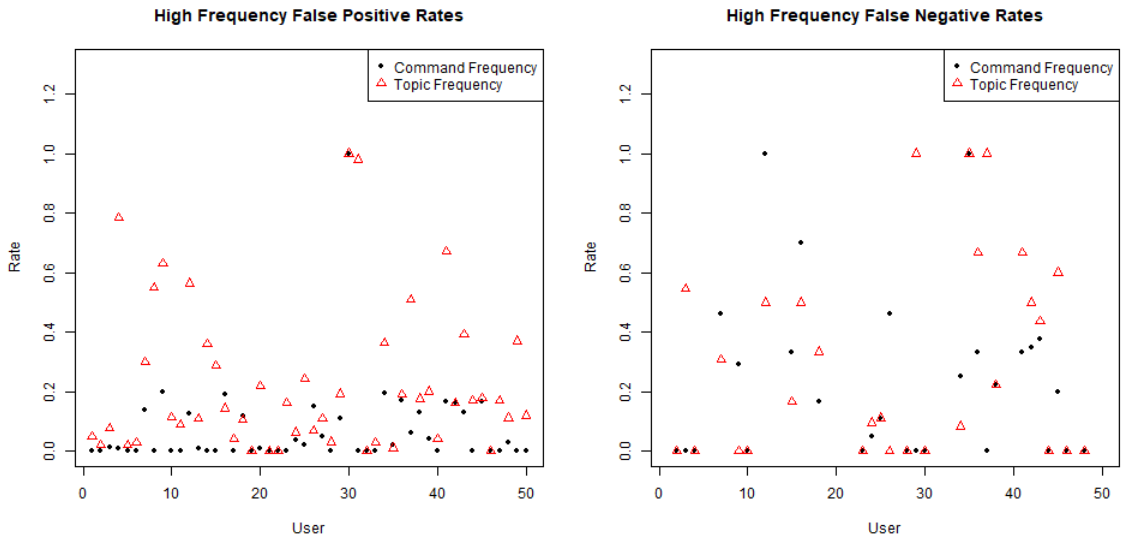
(user 30 and 31) had approximately 100% false positive rates when using the topic frequency method. See that user 30 has the same nearly 100% false positive rate for both methods. This is due to the fact that almost every block of topics was assigned to be a masquerader. For this user, a topic only had to occur 2 or more times to be considered a high frequency topic. This meant that almost every topic used by user 30 was a high frequency topic. It turns out that user 30 is the only user where such a low threshold was used and may indicate that this user had a number of sparse topics and thus small changes were mistaken for masqueraders. This can be seen in Figure 5.6a where the training, and most of the testing, blocks all belong to the same small number of commands and thus any changes from this seem to indicate a masquerader.

Figure 5.5b shows the comparison of the individual users false negative rates for the two methods. As with the false positive rates there are only a number of users which have higher rates for the command based or the topic based and all other users are about the same rate for both methods. This suggests that both methods do about the same for false negative rates. It is important to note that users 12 and 35 have nearly 100% false negative rate using the command based method and users 29, 35, and 37 have nearly 100% false negative rates for the topic based method. Looking at user 35, as both methods do poorly, I see that there is only one masquerade block, therefore, the entire false negative rate comes from mislabelling this one block. Moreover, most of the commands in this masquerade block belong to topic 23, 32 commands out of a possible 100, which is a topic that occurs in the training blocks with a frequency of 25%. Therefore, this block appears to be from the user instead of a masquerader. This can further be seen from Figure 5.6b as the one masquerader block has mostly the same commands as the training data. Only a few commands are new to this block of commands, but, as I are only considering high frequency topics and commands, these seem to not be enough of a difference to signal a masquerader.

Next, I compare the results from my high frequency topic model based method to the DDSGA method (see Table 2.1). I can see that the proposed method doesn't do as well as the DDSGA method. Lastly, I look at the cross validation of the high frequency topic method. As with the entire SEA dataset, I get a false positive of 22.44%. Similar false negative rates are also found (approx. 30%).

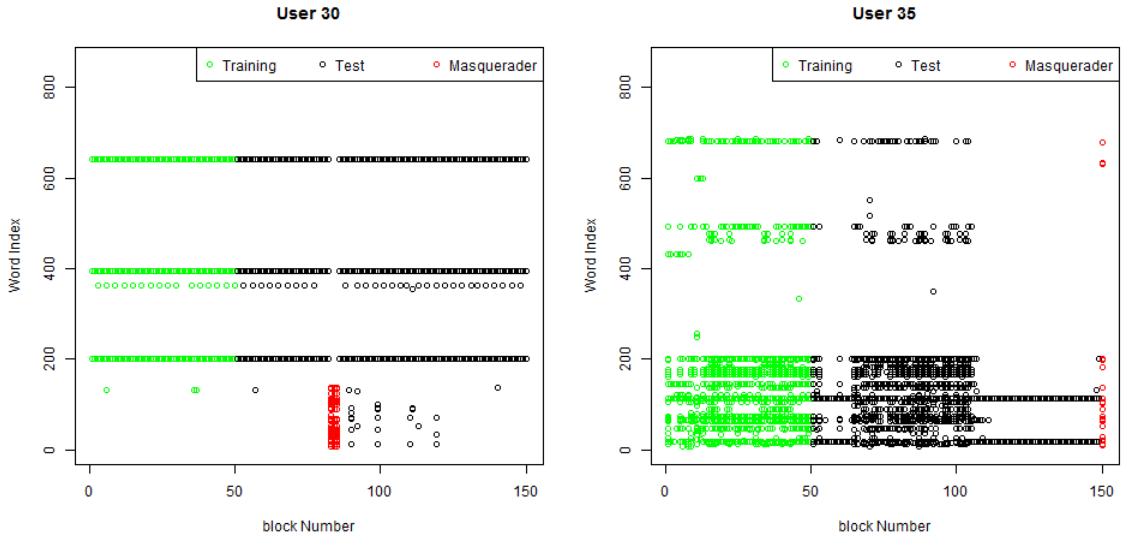
	TP	FP	TN	FN
High Frequency Commands	67.27	6.91	93.09	32.73
High Frequency Topics	69.88	22.44	77.56	30.12
Cross Validation	71.71	22.44	77.56	28.29

Table 5.3: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the high frequency topics method compared to using high frequency commands using the SEA dataset. Also included are the cross validation averaged rates.



(a) (b)

Figure 5.5: Comparison of false negative and false positive rates for individual users based on the high frequency methods using the SEA dataset.



(a) User with high FP rate. (b) User with high FN rate.

Figure 5.6: Closer look at command occurance by block for the users which have high FP and FN rates for all high frequency methods.

Greenberg Dataset

We have now repeated the analysis using the Greenberg dataset rather than the SEA dataset for the high frequency methods. Although neither of the high frequency topic based methods

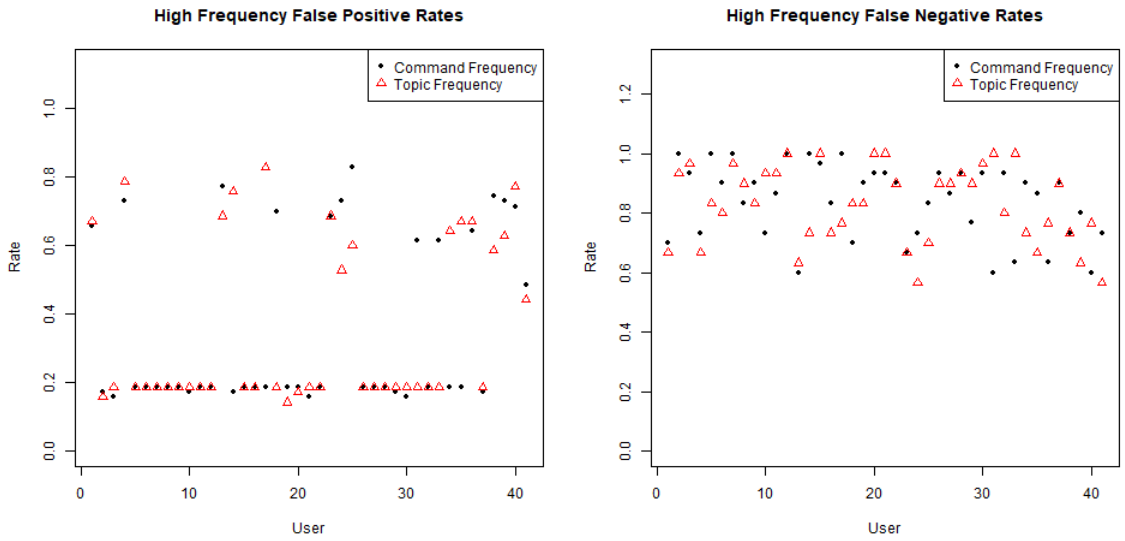
do particularly well at finding masqeraders (high false negative rates), from Table 5.4 I see that the average false positive rate is much lower for the topic based method than for the command based method. This, however, is achieved at the expense of the false negative rate which goes from 63% to 83%.

Next, I want to investigate individual users false negative and false positive rates to see which users are causing the changes in the average rates. For this I look at Figure 5.7. First, Figure 5.7a shows the individual users' false positive rates. The figure shows that about 2/3 of the users have a false positive rate at or just below 20% when using the topic based method and the command based method. The other users, for both methods, have FP rates between 60% and 80%. Second, Figure 5.7b shows the individual users' false negative rates. From this figure, I see that almost all of the users have rates above 60% when using the both the topic based and command based methods.

Next, I consider the cross validation of the high frewuency method to see how the method works with a subset of the Greenberg data. Table 5.4 shows that the CV values are on par with the command based method for the false negative rate and slightly imporoved for the false positives. In particular, the FP is around 22% compared to the 35% I achieved using the entire dataset, in addition, the FN is around 80% compared to 83% for the command based method and the topic based method on the entire dataset. The cross validation results indicate that the proposed high frequency method performs slightly better than the initial results suggested when considering false positives.

	TP	FP	TN	FN
High Frequency Commands	16.18	35.37	64.63	83.82
High Frequency Topics	17.15	35.85	64.14	82.85
Cross Validation	20.26	21.68	78.32	79.74

Table 5.4: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the high frequency topics method compared to using high frequency commands using the Greenberg dataset. Also included are the cross validation averaged rates.



(a) (b)

Figure 5.7: Comparison of false negative and false positive rates for individual users based on the high frequency methods using the Greenberg dataset.

5.3.3 Method 4: Low Frequency Topics

SEA Dataset

Similar to the high frequency topics method, both the command based and topic based methods have about the same average rates. This, combined with the reduction in dimensionality afforded by the topic method, indicates that the low frequency topics method does a slightly better job than the low frequency commands method.

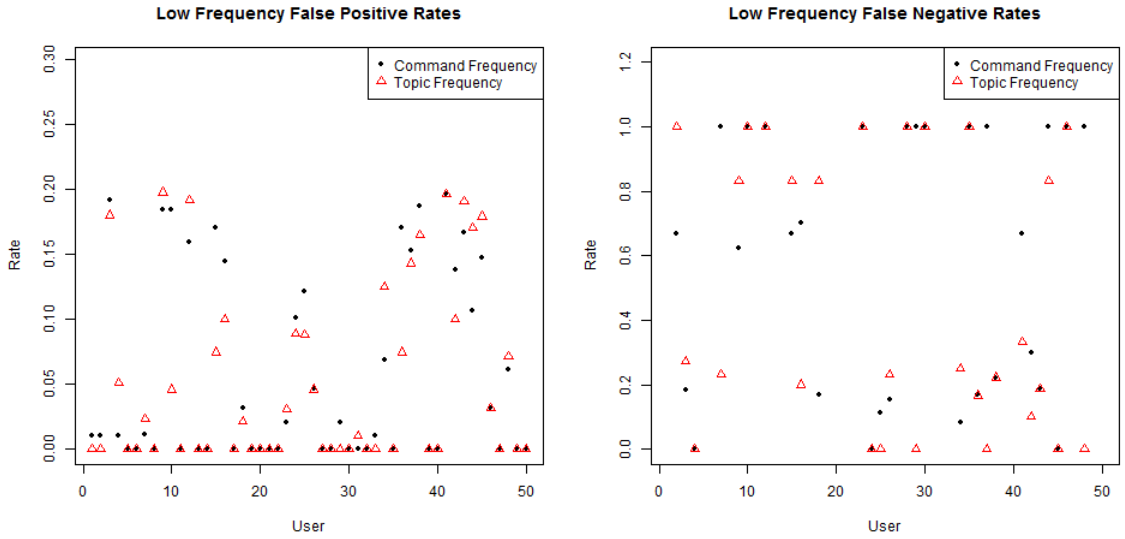
Next, I consider the individual users false positive and false negative rates (Figure 5.8). Both methods have a low false positive rate as seen in Figure 5.8a. All users have a false positive rate below 20% with about half having a 0% false positive rate. Additionally, when comparing the two methods, neither appears to have a better false positive rate for all the individuals. In fact, the rates are very similar for all users.

Next, I consider the false negative rates (see Figure 5.8b). Here I see that in about one-third of the users, both methods have an approximately 100% false negative rate. Also, of those users with approximately 100% false negative rates using the command frequency method, only a handful have a better rate using the topic frequency method, but, no user has a 100% false negative rate using the topic frequency method and a better rate using the command frequency method. All other users have similar rates using both methods. This provides further evidence that the low frequency topic method does slightly better than the low frequency command method.

Next, I compare my results to the DDSGA method (see Table 2.1). I can see that the proposed method doesn't do as well as the DDSGA method. Finally, I look at the cross validation of the low frequency method using a subset of the SEA dataset. The false positive rates for the cross validation are very similar to the method using the entire SEA dataset. As for the false negatives, I see that the cross validation has a lower average rate which indicates that the low frequency method applied to the entire SEA dataset may be an extreme case meaning that this method may work better, in terms of false negative rate, for other datasets.

	TP	FP	TN	FN
Low Frequency Commands	58.27	5.70	94.30	41.73
Low Frequency Topics	46.65	5.19	94.81	53.35
Cross Validation	66.91	5.06	94.94	33.09

Table 5.5: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the low frequency topics method compared to using low frequency commands using the SEA dataset. Also included are the cross validation averaged rates.



(a) (b)

Figure 5.8: False negative and false positive rates for individual users comparing the low frequency topics and low frequency commands methods using SEA dataset.

Greenberg Dataset

As with the low frequncy method on the SEA dataset, the average rates are similar for both the topic and command based methods using the Greenberg data. Therefore, the reduction in dimensionality afforded by the LDA topic model and the fact that my rates are on par

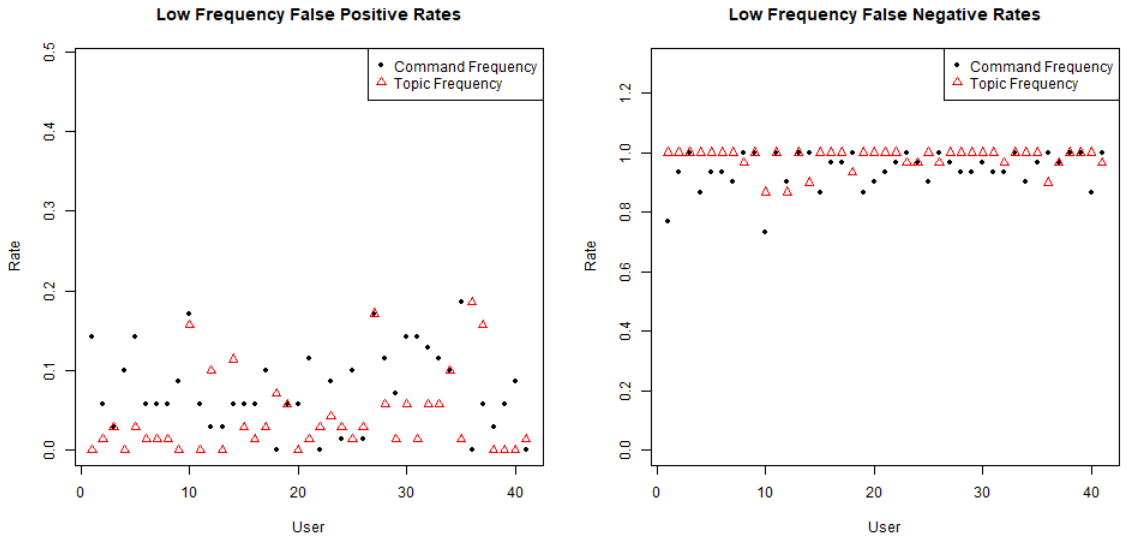
with the non-topic based method I conclude that the proposed novel intrusion detection method does a slightly better job at finding masqueraders.

Now, I considered the individual rates for each user (see Figure 5.9). First, Figure 5.9a shows the FP rates for each user. Here I see that the majority of users from both methods have a FP rate around 5%. All of the users FP rates for both methods were below 20%. Figure 5.9b shows the individual users FN rates. From here I see that most of the users have rates around 100% with slightly better rates for the command based method. This is as expected given the average rates.

Finally, I consider the cross validation of the topic based low frequency method using a subset of the Greenberg dataset. Table 5.6 shows that the false positive average rate is around 15% which is higher than I expected given the rate of the full dataset. However, the false negative has lowered to 80% from approximately 100%. Therefore, the cross validation achieved a slightly better result for the false negatives but at the cost of the false posiives.

	TP	FP	TN	FN
Low Frequency Commands	5.69	7.63	92.37	94.31
Low Frequency Topics	1.87	4.25	95.75	98.13
Cross Validation	18.76	14.51	85.49	81.24

Table 5.6: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the low frequency topics method compared to using low frequency commands using the Greenberg dataset. Also included are the cross validation averaged rates.



(a) (b)

Figure 5.9: False negative and false positive rates for individual users comparing the low frequency topics and low frequency commands methods using Greenberg dataset.

5.4 Method 5: Combination of Methods

Next I investigate combining the four methods above to try and better the true positive and negative rates using the SEA dataset. Table 5.7 shows the overall rates for the combined method plus the best individual method for comparison. I can see that the combined method does no better than a single individual method, but requires additional effort and thus additional computational time.

	TP	FP	TN	FN
PCA Topic Frequency	86.04	4.50	95.50	13.96
Combined	87.19	4.56	95.44	12.81

Table 5.7: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the combined method compared to using the best single method (PCA Topic Frequency) using the SEA dataset.

5.5 Method 6: Adaptive Naive Bayes

SEA Dataset

From the average rates shown in Table 5.8 there is a slight increase in the average false negative rates when comparing the topics and command based methods. The false positives also have a slight difference with the proposed method having almost 1% less. However, the dimensionality and compute time are reduced for the topic based method thus providing evidence that the method proposed here using topic frequencies rather than command frequencies has benefits.

Next, consider the individual user's false negative and false positive rates rather than the averaged rates (see Figure 5.10). First, looking at Figure 5.10a, the individual false positives, the rates are similar for both methods. There are a number of users which have slightly higher rates for the topic based method and similarly for the command based method but not overly large compared to their counterpart. Importantly, one user, user 13, has high false positive rates using both methods. This indicates that for this user the adaptive naive Bayes method is overfitting. From Figure 5.11 the testing data and the training data is quite different, however, there are no masqueraders. This difference is due to changes in the users behaviour. My method, however, captures these changes and wrongly contributes them to a masquerader.

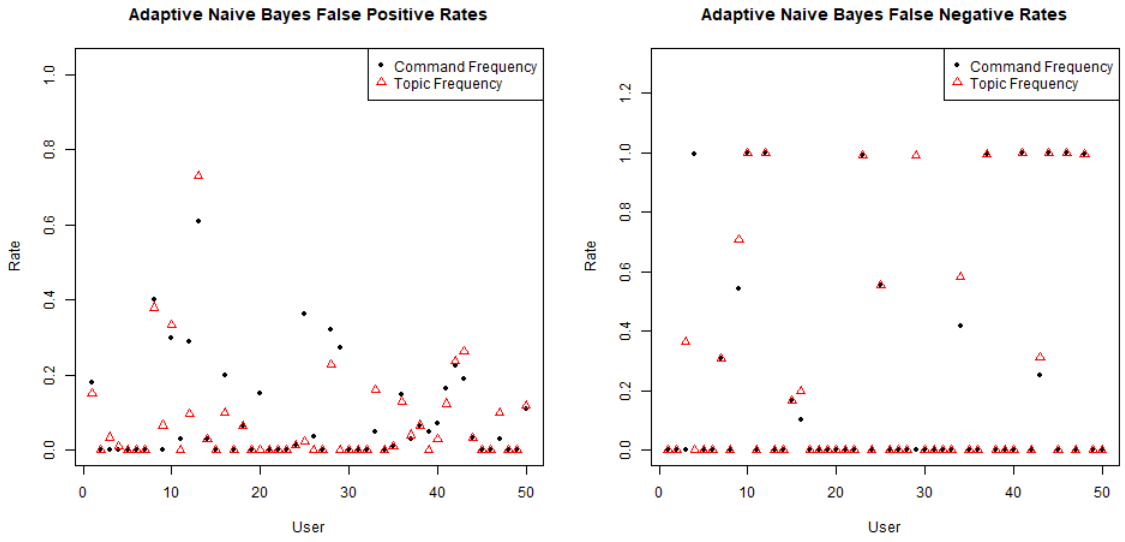
Second, from Figure 5.10b the individual false negatives are essentially the same for all users, with the exception of three users who have higher false negative rates for the topic based method. It is also important to note that the users with less than three masquerade blocks will most likely be mislabelled as the adaptive naive Bayes method requires three consecutive blocks to be labelled as doubtful before the blocks are labelled as masqueraders.

Next, compare the new topic model based method the DDSGA method. From Table 2.1 the DDSGA method achieves a false positive rate of 3.4 where as the proposed method has a false positive rate of 7.13. Additionally, the proposed method achieves much worse false negative rates.

Finally, compare the proposed method using the entire SEA dataset to using a cross validation on a subset of the SEA dataset. In the cross validation case, I achieve a false positive rate of approximately 4% compared to the 7% I got using the entire SEA dataset. This slight diiference in false positive rate, however, is negligible. This was achieved at the expense of the false negative rate which doubled. However, this high false negative rate is probably due to the fact that adaptive naive Bayes requires consecutive blocks to be masqueraders which is built into the SEA dataset, but not necessarily found in the cross validation test set.

	TP	FP	TN	FN
Ad. Naive Bayes Commands	77.40	8.86	91.14	22.60
Ad. Naive Bayes Topics	75.69	7.13	92.87	24.31
Cross Validation	49.76	4.29	95.71	50.24

Table 5.8: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the adaptive naive Bayes for topics method compared to using adaptive naive Bayes for commands method using the SEA dataset.



(a) (b)

Figure 5.10: False negative and false positive rates for individual users comparing the adaptive naive Bayes topics and adaptive naive Bayes commands methods using the SEA dataset.

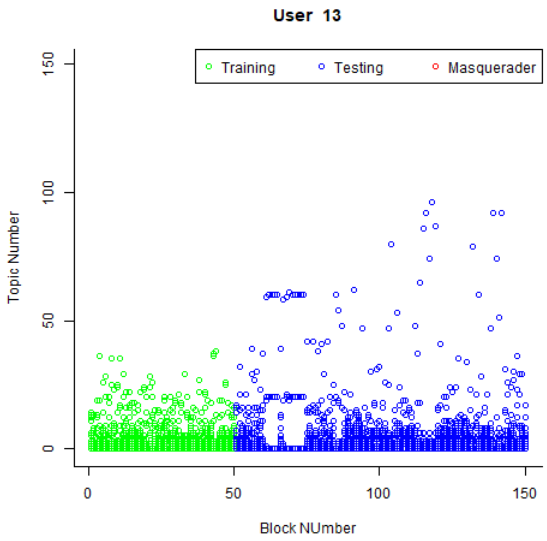


Figure 5.11: Topic occurance by block for user 13 which has a high FP rate.

Greenberg Dataset

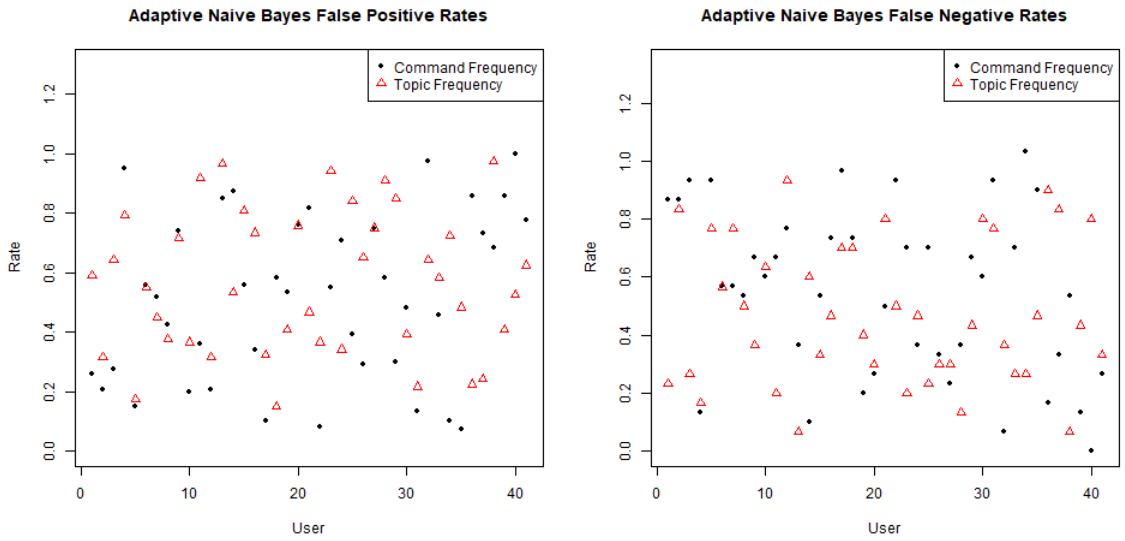
We repeat the analysis of the adaptive naive Bayes method using the Greenberg dataset. Table 5.9 shows the average rates which are consistent between the topic and command based methods. Both methods were only able to identify a masquerader correctly about half of the time. There is no obvious benefit to using one method over the other, other than the reduction in compute time from the reduction in dimensionality due to using topic modelling for the topic based method over the command based method.

Next, consider individual users FN and FP rates (see Figure 5.12). For both the false negative and false positive rates about half the users have good rates and half have bad rates which is why the average rate is approximately 50%. It is really a 'coin toss' to see if the masqeurader is detected correctly or not.

Finally, consider the cross validation of the topic based adaptive naive Bayes method using a subset of the Greenberg dataset. Table 5.9 shows that the cross validation average rates are on par with the full dataset results, of a nearly 50% detection rate.

	TP	FP	TN	FN
Ad. Naive Bayes Commands	45.20	51.36	48.64	54.80
Ad. Naive Bayes Topics	52.52	56.22	43.78	47.48
Cross Validation	44.40	47.53	52.47	55.60

Table 5.9: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the adaptive naive Bayes for topics method compared to using adaptive naive Bayes for commands method using the Greenberg dataset.



(a) (b)

Figure 5.12: False negative and false positive rates for individual users comparing the adaptive naive Bayes topics and adaptive naive Bayes commands methods using the Greenberg dataset.

5.6 Modified Block Size

To further investigate the proposed methods I use the dataset with large block sizes and make new datasets with smaller block sizes. That is, originally block sizes were x now I tried sizes $x/2$, $x/4$ and $x/10$. Similarly, I use my dataset with small block sizes and create a new dataset with larger blocks, originally blocks were of size y and I tried blocks of size $2y$ and $5y$. I use these modified block sizes to test my methods to see which works better, a smaller number of larger blocks or a larger number of smaller blocks. In this way, I get a better understanding of what type of data is needed for my methods to perform well. I only use these new datasets on the method which performed the best for the original dataset.

Here look at what happens to the average rates when the block sizes are modified. To get a better understanding of what block size should be used to get the best average rates. First, consider the SEA dataset which originally had blocks of 100 commands. I modified this by considering blocks of size 50, 25, and 10. I then use the PCA with topic frequencies method as this method had the best results for the original dataset. Looking at Table 5.10 having more smaller blocks results in higher FP and FN average rates. As the block sizes got smaller the average false positive rate increased slightly and the average false negative rate increased considerably.

Block Size	TP	FP	TN	FN
10	59.22	6.97	93.03	40.78
25	64.61	7.33	92.67	35.39
50	71.49	6.92	93.08	28.51
100	86.04	4.50	95.50	13.96

Table 5.10: True Positive, false positive, true negative, and false negative rates averaged over all 50 users for the Principal Component Analysis with Topic Frequencies method using the SEA dataset with varying block sizes.

Second, I use the Greenberg dataset which originally had blocks of size 10 and modify the data to have blocks of size 2, 20, 50, and 100. I then used the adaptive naive Bayes topics method and the PCA topic frequency method as these provided the best results for the original dataset. Table 5.11 shows the averaged rates for the varying block sizes using the adaptive naive Bayes method. From this table, I see that the average FP rates seems to be unaffected by the block size, but the false negative average rate appears to be decreasing as the block size increases.

Table 5.12 shows the average rates using the PCA topic frequency method. From this table I see a slight increase in false negatives as the block size gets really small, 2, and a decrease as the block size gets really big, 100. The false positives, on the other hand, seem to remain the same with a slight decrease for a block size of 2.

Block Size	TP	FP	TN	FN
2	52.93	54.10	45.90	47.07
10	52.52	56.22	43.78	47.48
20	52.03	57.56	42.44	47.97
50	55.20	57.52	42.48	44.80
100	56.10	56.30	43.70	43.90

Table 5.11: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the adaptive naive Bayes topic method using the Greenberg dataset with varying block sizes.

Block Size	TP	FP	TN	FN
2	10	9.57	90.43	90
10	17.26	16.10	83.90	82.74
20	15.61	18.05	81.95	84.39
50	14.80	16.67	83.33	85.20
100	22.85	16.06	83.94	77.15

Table 5.12: True Positive, false positive, true negative, and false negative rates averaged over all 41 users for the Principal Component Analysis with Topic Frequencies method using the Greenberg dataset with varying block sizes.

5.7 Computational Effort

We finally look at the actual run times of the SEA dataset for all six of my methods compared to their original non-topic based counter-parts (see Figure 5.13).

First, for the PCA based methods I extended the original method in two ways so the command based counter-part compute times for both methods is the same. Next, I see that both of my proposed PCA-based extensions run in under a second where as the original method takes nearly 37 seconds. Therefore, my methods 1 and 2 run considerably faster. This is due to the reduction in dimensionality afforded by doing topic modelling.

Second, I look at method 3, the high frequency method, and method 4, the low frequency method. For the high frequency commands method using the SEA dataset I achieve a runtime of 4.09 seconds compared to the proposed method which has a better 3.11 seconds. Although that may seem like an insignificant difference, real-world data will be much larger and have higher dimensionality and thus this small difference will become a larger difference as the compute times increase. Therefore, there appears to be some evidence that the proposed method runs faster and is therefore better. Next, the command based low frequency method. In this case the proposed method runs considerably slower than the command based counter-part. This provides evidence that the proposed method may not be as good as the original command based one.

Third, I look at the combination of the first four methods and compare it to the runtime of the single best individual method (PCA Frequency). The runtime for the combined method is the combined runtime of all the methods plus the runtime to compare the methods and find the consensus. This means that the runtime for this combined method is always going to be more than the best individual method and in this case it is considerably more and the low and high frequency methods each take five times as long as the PCA Frequency based topic model. Therefore, I have evidence that the combined method doesn't do as well as the best individual method.

Finally, I consider the runtime of the Adaptive Naive Bayes method. Here I see that my topic based extension runs in 6.6 seconds where as the command based one takes a whopping 127 seconds. This is a huge improvement for the proposed method and provides evidence that the proposed method is superior.

Out of all of the methods I find that the PCA Topic Frequency based method and the Low Frequency Command based method have the fastest runtimes and the Adaptive Naive Bayes has the best improvement over the command based counter-part.

It is important to note that the run times are just for the given methods and an initial run time of 905 seconds for the LDA topic modeling needs to be taken into account as well.

	Command Based	Topic Based
PCA Prob	36.96	0.95
PCA Frequencies	36.96	0.58
High Frequency	4.09	3.11
Low Frequency	0.23	3.24
Combined	-	All+0.36
Ad. Naive Bayes	126.83	6.6

Table 5.13: Comparison of compute times (in seconds) for the command based and topic based models using the SEA dataset.

Chapter 6

Conclusions and Future Work

This project considered intrusion detection techniques based on topic modelling to extend current intrusion detection techniques. Namely, I extended the PCA based technique proposed by Wang [25] in two ways. First, I used topic distributions rather than command frequencies when doing PCA. Second, I used topic frequencies within a block for the PCA. Additionally, several approaches were modified so that the high [24] and low [22] frequencies of commands I instead used frequencies of topics. In this way, I created two new methods: high frequency topics and low frequency topics. Finally, I modified the adaptive naive Bayes technique proposed by Dash et al. [10] by using topic frequencies rather than command frequencies.

The methods were implemented on the Unix command data from SEA and from Greenberg. Experimental results indicated that all of the methods were on par with the command based counterparts in terms of overall average and individual user FP and FN rates. Combining these findings with the lower computational complexity achieved by the dimension reduction of topic modelling I found that my topic modelling based methodologies work as well as or better than the command based methodologies.

However, no one method seems to provide the best results for both datasets. For instance, the PCA topic frequency method and the combined method performed well on the SEA dataset, although the combined method takes much longer to compute thus the PCA topic frequency method is prefered. For the Greenberg dataset, on the other hand, the low frequency method works well for false positives, but has nearly a 100% false negative rate. Both the PCA based methods work relatively well for the Greenberg dataset, but still have a high false negative. In the end, we recommend that following the steps for comparison of methods, as was done here, is required before using any method in a practical setting.

When considering why the methods appear to work well on the SEA dataset, but poorly on the Greenberg dataset, the approach I took when constructing Figure 5.1 sheds some light. The Mahalanobis distance for each test block of topic frequencies from the average training topic frequencies. Using the SEA dataset, most of the masquerader blocks truely

are different and far from the training data and, usually, the testing data is close to the training data. For those users with dissimilar test blocks compared to the training blocks I would expect to get false positives and similarly for masquerader blocks which are similar to the training data giving false negative. These are rare in the SEA dataset but common in the Greenberg dataset. Furthermore, the Greenberg dataset doesn't appear to have many blocks, masquerader or otherwise, which are far from the training data. All blocks appear to be of a similar distance from the training data.

Since the success of the intrusion detection technique is dependent on the dataset, a company wishing to perform intrusion detection should follow these steps:(1) First, the company needs to choose 2-3 models that they wish to use. These models will then be tested using training data to see which one works the best for the given dataset. (2) The company then needs to get some clean data to test the models with. This data is typically taken from the previous day(s). (3) A block size needs to be chosen. The company can choose any block size, but I suggest that they try blocks of size 50, 25, and 10. Each of these block sizes will be tested with each of the chosen methods and then the best block size will be used for future testing. (4) The clean data needs to be split into training and testing data. Thus, the company needs to decide on a number of blocks to train each method on and the rest of the blocks will be used to test the methods. (5) The number of times a command occurs within a block needs to be calculated. (6) The clean testing data then needs to be interspersed with simulated masquerade data. This can be done in a number of ways. First, the data can be split into users and potential masqueraders. Then, the Mahalanobis distance will be calculated from the mean of each user to the mean of each potential masquerader and the furthest masqerader will be chosen. Some data from the chosen masquerader is randomly inserted into the users test data. Second, the company could instead use the Mahalanobis distance to calculate the distance between the training data and each command and then assume this distribution of command counts is normal and get simulated data blocks of command counts such that the simulated blocks have around 30% of their commands from commands which have large Mahalanobis distances. (7) Run each IDS which was chosen in step 1 with the training and testing data with block sizes chosen in step 3 to see which IDS method and block size combination works best with the data. (8) Select the best IDS and block size such that the average FN and FP rates are lowest. (9) Once and intrusion detection system and block size has been selected, new data can be checked for masqueraders as soon as it occurs.

To extend this project one could consider first grouping the users in a reasonable manner. This could include grouping users based on similarity of commands. Therefore, although the methods proposed in this project have been done at an individual user level (see Algorithm 5 and 7), it is possible to also do the topic modelling at a user group/cluster level (see Algorithm 6 and 8).

Another possible extension to this project is to use Markov Random Field LDA topic modelling. As mentioned previously, one of the key assumptions of LDA is the *bag-of-words* approach to word (in this case command) selection. However, in the context of masquerade detection, this assumption may not be appropriate. This is due to the fact that commands are often correlated to one another and occur close together. For instance, if a user makes a directory (mkdir) it is likely that shortly afterwards the user will change to that newly made directory (cd). Therefore, the basic LDA approach needs to be extended to account for correlations between words. In 2005, Metzler and Croft [18] proposed a model to incorporate term dependencies into topic modeling using Markov Random Field (MRF). Similarly, in 2015 Pengtao Xie et. al. [26] proposed a MRF regularized LDA method which extends the LDA method to include a matrix of word correlations and uses these correlations to encourage similar words to belong to the same topic. These two methods could be used as the topic modelling step in my proposed methods which would allow for correlation between commands to be taken into account.

In conclusion, my six topic model based intrusion detection methodologies do as well or better than their current command based counter-parts. In addition, using topic models allows for a reduction in dimensionality which leads to a reduction in compute time and thus provides more timely real-world use. Out of the six methods proposed in this project the PCA Topic Frequency based method provided the best overall rates and compute time reduction for the SEA data and the low frequency based method works the best for the Greenberg data. The Greenberg dataset had no statistical power for any of the methods used for either the command or the topic based methods. I have also outlined the methodology needed for a company to run intrusion detection using one of my systems and provided some guidelines on block size vs number of blocks needed to get good results.

Bibliography

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] D. M. Blei. Surveying a suite of algorithms that offer a solution to managing large document archives. probabilistic topic models. *Communications of the acm*, 55(4):77–84, 2012.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[4] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, and S. Gombault. Efficient intrusion detection using principal component analysis. In *3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe, France*, pages 381–395, 2004.

[5] B. Camiña, R. Monroy, L. A. Trejo, and E. Sánchez. Towards building a masquerade detection method based on user file system navigation. In *Mexican International Conference on Artificial Intelligence*, pages 174–186. Springer, 2011.

[6] J. B. Camiña, J. Rodríguez, and R. Monroy. Towards a masquerade detection system based on userâĂŹs tasks. In *International Workshop on Recent Advances in Intrusion Detection*, pages 447–465. Springer, 2014.

[7] X. Cao, B. Chen, H. Li, and Y. Fu. Packet header anomaly detection using bayesian topic models. 2016.

[8] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[9] C. Cramer and L. Carin. Bayesian topic models for describing computer network behaviors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1888–1891. IEEE, 2011.

[10] S. K. Dash, K. S. Reddy, and A. K. Pujari. Adaptive naive bayes method for masquerade detection. *Security and Communication Networks*, 4(4):410–417, 2011.

[11] W. DuMouchel. Computer intrusion detection based on bayes factors for comparing command transition probabilities, 1999.

[12] J. Eng. Ubiquiti networks says it was victim of $47 million cyber scam. 2015.

[13] S. Greenberg. Using unix: Collected traces of 168 users. 1988.

[14] J. Huang, Z. Kalbarczyk, and D. M. Nicol. Knowledge discovery from big data for intrusion detection using lda. In *IEEE International Congress on Big Data*, pages 760–761. IEEE, 2014.

[15] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[16] R. Kemmerer and G. Vigna. Intrusion detection: a brief history and overview. *IEEE Security and Privacy Magazine*, 2002.

[17] H. A. Kholidy, F. Baiardi, and S. Hariri. Ddsga: A data-driven semi-global alignment approach for detecting masquerade attacks. *IEEE Transactions on Dependable and Secure Computing*, 12(2):164–178, 2015.

[18] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM, 2005.

[19] J. Rodríguez, L. Cañete, R. Monroy, and M. A. Medina-Pérez. Experimenting with masquerade detection via user task usage. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(4):771–784, 2017.

[20] M. Schonlau. Masquerading user data.

[21] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi. Computer intrusion: Detecting masquerades. *Statistical Science*, 16:1–17, 2001.

[22] M. Schonlau and M. Theus. Detecting masquerades in intrusion detection based on unpopular commands. *Information Processing Letters*, 76(1):33–38, 2000.

[23] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

[24] M. D. Wan, H.-C. Wu, Y.-W. Kuo, J. Marshall, and S.-H. S. Huang. Detecting masqueraders using high frequency commands as signatures. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 596–=–601. IEEE, 2008.

[25] W. Wang, X. Guan, and X. Zhang. A novel intrusion detection method based on principle component analysis in computer security. In *International Symposium on Neural Networks*, pages 657–662. Springer, 2004.

[26] P. Xie, D. Yang, and E. P. Xing. Incorporating word correlation knowledge into topic modeling. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

Appendix A

Method Pseudocode

This section provides pseudocode for the methods used in this project. For all of the methods,

1. Let **M** be the topic model created using training data from all users.
2. Let **p** be the list of matrices of topic probability distributions using **M** such that **p[i]** is the m-by-n matrix for user i, where m is the number of data blocks and n is the number of topics in **M**.

p[i]=

prob block 1 belongs in topic 1	prob block 2 belongs in topic 1	...	prob block m belongs in topic 1
...			
prob block 1 belongs in topic n	prob block 2 belongs in topic n	...	prob block m belongs in topic n

3. Let **c** be the vector of associated topics for each command as chosen using the topic distribution of the commands using **M**.
4. Let **C** be the list of matrices of the frequency of topics, using **c**, within a block of commands such that **C[i]** is the m-by-n matrix for user i.

C[i]=

freq of topic 1 in block 1	freq of topic 1 in block 2	...	freq of topic 1 in block m
...			
freq of topic n in block 1	freq of topic n in block 2	...	freq of topic n in block m

A.1 Method 1 and 2: PCA LDA

Algorithm 1 Pseudocode for Principal Component Analysis based methods

For method 1 data is **p[i]**
For method 2 data is **C[i]**

Divide data by the number of commands per block
Split the modified data into training, D_i, and testing, d_i
Let τ be a vector of possible thresholds

Calculate mean(D_i) and the eigenpairs of the sample covariance of D_i
Keep eigenvectors which explain 99.9% of variance, denoted **U**
for all d_i and τ pairs **do**
    Calculate the mean adjusted d_i (denoted ϕ),
    the projection of d_i onto the subspace given by **U** (denoted **y**),
    $\phi_f = \mathbf{U}\mathbf{y}$,
    and the squared Euclidean distance between ϕ and ϕ_f (ϵ)
    Compare ϵ and τ to determine if d_i is a masquerader
Select best τ for each d_i

A.2 Method 3 and 4: High and Low Frequency Topics

Algorithm 2 Pseudocode for method 3 and 4

Split **C[i]** into training and testing blocks
Calculate average freq of topics in training blocks by

$$F = \sum(\mathbf{C[i][,1:ntrain]})/\sum(\mathbf{C[i][,1:ntrain]} > 0)$$

Calculate probability of topic occuring in block by dividing F by length of block
Sort probability of topics occuring by highest (lowest) probability
if method 3 **then**
    Only keep k highest probability of occuring topics
else
    Only keep k lowest non zero probability of occuring topics
for all test blocks in **C[i]** **do**
    Calculate the probability of the k kept topics within the test block
    Test for masquerader by comparing test block probability to training block probability for k kept topics

Appendix B

Threshold Validation

Algorithm 3 Pseudocode for method validation

Let D_i be the set of training data for user i
Let d be the set of testing data for all users
Let d_i be the set of testing data for user i
Let M_j^k be the vector of masquerader indicators for threshold j with data k where 0 indicates the block of data is from the user and 1 indicates the data block is a masquerader
Let $\{M^k\}|_m$ be a set of m masquerader indicator vectors for data k

function MASQUERADE INDICATOR CONSENSUS($\{M^k\}|_m$)
  for all i = 1 ... length(M^k) **do**
    if $\sum M^k[i] \geq 0.5$m **then** return 1
    else return 0

for all Users i **do**
  for j = 1 ... n **do**
    Let R_u be N^{train} randomly chosen blocks of data from D_i
    Let R_n be the remaining blocks of data from D_i
    for k = 1 ... N **do**
      Let **r** be N^{test} randomly chosen blocks of data from d
      Let **T** be the set of **r** and R_n
      Run method using training data R_u and testing data **T** to find threshold ω_k
      Using R_u, d_i and ω_k get $M_{\omega_k}^{d_i}$
    function MASQUERADE INDICATOR CONSENSUS($\{M^{d_i}\}|_N$)
  function MASQUERADE INDICATOR CONSENSUS($\{M^{d_i}\}|_n$)
  Calculate FN, FP, TN, and TP

Appendix C

Assignment of LDA topics to commands

Algorithm 4 Assignment of topics to individual commands

Let **C** be the vector of commands
Let $\mathbf{P}_{ij}$ be the probability that command **C[i]** belongs in topic j
Let **T** be the vector of chosen topics such that **T[i]** is the topic for **C[i]**
function BEST TOPIC(i)
    T[i] = $\max_j \mathrm{P}_{ij}$
function TOPIC WEIGHTS(i)
    T[i] = Select topic with probability P_i
function COIN FLIP(i)
    Let τ be the probability threshold
    Let $\mathbf{t}_{ij}$ be the topics such that $\mathbf{P}_{ij} \geq \tau$
    Let **n** be length($\mathbf{t}_i$)
    T[i] = Select topic from $\mathbf{t}_i$ with probability $\frac{1}{n}$

Algorithm 5 Topic Model Individual Command Frequencies without Clusters

Perform topic modeling using training data from ALL users
Assign individual commands to topics via Algorithm 4
for all users **do**
    Calculate expected frequency of topics using only the 50 training blocks for the user
    for all test blocks **do**
        Calculate frequency of topics
        Test if masquerader

Algorithm 6 Topic Model Individual Command Frequencies with Clusters

Perform topic modeling using training data from ALL users
Assign individual commands to topics via Algorithm 4
T = (number of training blocks)x(block size) matrix of topic numbers such that each row corresponds to one block of commands within the training data
Cluster users based on topic model of training data
for all clusters **do**
    N_{ijk} = number of times topic i appears in training block j within cluster k as seen from **T**
    $N_{ik} = \sum_j N_{ijk}$
    p_{ik} = probability of topic i appearing in cluster $k = N_{ik}/N_k$
for all users **do**
    N_{ijl} = number of times topic i appears in training block j within user l as seen from **T**
    $N_{il} = \sum_j N_{ijl}$
    p_{il} = probability of topic i appearing in user $l = N_{il}/N_l$
    for all test blocks **do**
        Calculate frequency of topics
        Test if masquerader using expected frequency determined from individual user
        if Test block not from user as indicated from individual masquerader test **then**
            for users cluster **do**
                Test if masquerader using expected frequency determined from cluster

Appendix D

Assignment of LDA topics to command blocks

Algorithm 7 Topic Model Blocks of Commands without Clusters

```
1: Perform topic modeling using training data from ALL users
2: for all training blocks do Get probabilities associated with each topic
3: for all users do
4:     function Expected topic probabilities(users training data prob.)
5:         Each training block corresponds to a vector of probabilities
6:         Expected topic probability = average topic probability over all training blocks
7:     for all test blocks do
8:         Calculate probabilities for the test block belonging to each topic
9:         Test if masquerader
```

Algorithm 8 Topic Model Blocks of Commands with Clusters

```
1: Perform topic modeling using training data from ALL users
2: for all training blocks do Get probabilities associated with each topic
3: Cluster users
4: for all clusters do
5:     function Expected topic probabilities(cluster training prob.)
6:         Each training block corresponds to a vector of probabilities
7:         Expected topic probability = average topic probability over all training blocks
8: for all users do
9:     function Expected topic probabilities(users training prob.)
10:    for all test blocks do
11:        Calculate probabilities for the test block belonging to each topic
12:        Test if masquerader using expected probability determined from individual user
13:        if Test block not from user as indicated from individual masquerader test then
14:            for users cluster do
15:                Test if masquerader using expected probability determined from cluster
```

Appendix E

Threshold

User	PCA Distance		Num Topics		Low Freq (f_L)
	Topic Freq	Probability	High Freq (N_H)	Low Freq (n_L)	
1	0.131	1e-05	56	1	0.01
2	0.009	1e-05	39	1	0.01
3	0.002	1e-05	39	37	0.01
4	0.016	1e-05	42	14	0.01
5	0.005	1e-05	47	1	0.01
6	0.009	1e-05	33	1	0.01
7	0.004	1e-05	39	5	0.01
8	0.018	1e-05	45	1	0.01
9	0.014	1e-05	47	11	0.01
10	0.149	6e-05	42	5	0.01
11	0.215	8e-05	55	1	0.01
12	0.018	3e-05	51	13	0.01
13	0.323	5e-05	54	1	0.01
14	0.035	1e-05	44	1	0.01
15	0.003	1e-05	56	18	0.01
16	0.155	7e-05	49	15	0.01
17	0.013	1e-05	54	1	0.01
18	0.016	1e-05	33	14	0.01
19	0.022	3e-05	51	1	0.01
20	0.806	9e-05	54	1	0.01
21	0.056	3e-05	33	1	0.01
22	0.006	1e-05	52	1	0.01
23	0.097	3e-05	56	16	0.01
24	0.008	1e-05	46	21	0.01
25	0.007	1e-05	47	35	0.01
26	0.015	3e-05	54	3	0.01
27	0.781	4e-05	54	1	0.01
28	0.019	3e-05	41	1	0.01
29	0.033	1e-05	37	1	0.01
30	0.014	1e-05	2	1	0.01
31	0.004	1e-05	33	1	0.01

32	0.025	3e-05	41	1	0.01
33	0.626	7e-05	43	1	0.01
34	0.002	1e-05	50	32	0.01
35	0.041	1e-05	36	1	0.01
36	0.02	1e-05	35	4	0.01
37	0.002	1e-05	47	23	0.01
38	0.007	1e-05	48	17	0.01
39	0.987	6e-05	49	1	0.01
40	0.201	3e-05	50	1	0.01
41	0.006	1e-05	44	19	0.01
42	0.004	1e-05	47	12	0.01
43	0.006	1e-05	49	23	0.01
44	0.028	1e-05	45	25	0.01
45	0.01	1e-05	34	9	0.01
46	0.001	1e-05	33	29	0.01
47	0.081	2e-05	27	1	0.01
48	0.024	1e-05	51	2	0.01
49	0.13	3e-05	55	1	0.01
50	0.025	1e-04	51	1	0.01

Table E.1: Optimal thresholds for the methods proposed in this project. For each of the two principal component analysis (PCA) based methods one threshold is used, namely the PCA distance. Both of the topic frequency based methods require a threshold for the number of topics used to indicate that a block is either high or low frequency. The low frequency topics method requires an additional threshold for the number of topics which have a within training block frequency of topics below this threshold. This threshold is present in the high frequency command method as well, but is set to zero and the threshold is for above zero rather than below.

Appendix F

Mahalanobis Distance

User	Average Test Distance	Masquerader Distance	Masquerader Distance from Avg
2	190899	73740726 , 6021310 , 260342238	73549827 , 5830411 , 260151339
3	203886	2276377 , 9201376 , 2275291 , 2324725 , 841244 , 10299454 , 2537689 , 62228 , 47099506 , 55372776 , 1259503	2072491 , 8997490 , 2071405 , 2120839 , 637358 , 10095568 , 2333803 , 141658 , 46895620 , 55168890 , 1055617
4	95739	76240708 , 14880304	76144969 , 14784565
7	797720	6677382 , 3731425 , 3781733 , 39345122 , 177798222 , 14666808 , 4762228 , 27796669 , 81888841 , 82565635 , 7750985 , 4349511 , 4000654	5879662 , 2933705 , 2984013 , 38547402 , 177000502 , 13869088 , 3964508 , 26998949 , 81091121 , 81767915 , 6953265 , 3551791 , 3202934
9	916709	2252771 , 1968574 , 1960816 , 2069811 , 1968574 , 1960816 , 2035997 , 1960816 , 1238403 , 3286050 , 4841856 , 5163817 , 16162651 , 10291829 , 784016 , 1066469 , 2829757 , 385046 , 321701 , 702255 , 2194344 , 1552062 , 2877503 , 914501	1336062 , 1051865 , 1044107 , 1153102 , 1051865 , 1044107 , 1119288 , 1044107 , 321694 , 2369341 , 3925147 , 4247108 , 15245942 , 9375120 , 132693 , 149760 , 1913048 , 531663 , 595008 , 214454 , 1277635 , 635353 , 1960794 , 2208
10	13151766	1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692 , 1025012692	1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926 , 1011860926
12	1603555	1242 , 22345 , 2423 , 24180 , 35436 , 1228	1602313 , 1581210 , 1601132 , 1579375 , 1568119 , 1602327

15	769759	11771483 , 11687308 , 10090487 , 12673395 , 5296277 , 572249	11001724 , 10917549 , 9320728 , 11903636 , 4526518 , 197510
16	10438616	2789617 , 227726 , 1048513 , 33700143 , 70918378 , 2686632 , 12413500 , 4497578 , 91812736 , 49736727	7648999 , 10210890 , 9390103 , 23261527 , 60479762 , 7751984 , 1974884 , 5941038 , 81374120 , 39298111
18	3516746	15352278 , 14789060 , 20161321 , 10692047 , 23587536 , 20384175	11835532 , 11272314 , 16644575 , 7175301 , 20070790 , 16867429
23	428604	1018563603	1018134999
24	187024	837261104 , 377707425 , 245667815 , 164741024 , 553520054 , 579520054 , 572920054 , 579520054 , 579520054 , 506588412 , 354312150 , 313032844 , 537920054 , 508538128 , 249854654 , 769619486 , 514510313 , 815814860 , 700788246 , 962588412 , 550286458	837074080 , 377520401 , 245480791 , 164554000 , 553333030 , 579333030 , 572733030 , 579333030 , 579333030 , 506401388 , 354125126 , 312845820 , 537733030 , 508351104 , 249667630 , 769432462 , 514323289 , 815627836 , 700601222 , 962401388 , 550099434
25	35248	12403540 , 45031311 , 354061291 , 2626861 , 108131997 , 70879201 , 108131997 , 103089484 , 85080688	12368292 , 44996063 , 354026043 , 2591613 , 108096749 , 70843953 , 108096749 , 103054236 , 85045440
26	2848132	96453082 , 11203243 , 10881802 , 26113627 , 10674307 , 6629469 , 72482198 , 81755614 , 35301082 , 41369965 , 40624035 , 56336773 , 34490616	93604950 , 8355111 , 8033670 , 23265495 , 7826175 , 3781337 , 69634066 , 78907482 , 32452950 , 38521833 , 37775903 , 53488641 , 31642484
28	1773481	23780470 , 24225495 , 22434680	22006989 , 22452014 , 20661199
29	14974173	35123560	20149387
30	257998	309540533 , 359213623 , 347013623	309282535 , 358955625 , 346755625
34	269131	106786344 , 111227023 , 90392129 , 115293105 , 106081465 , 4173038 , 4687338 , 263592 , 1081895 , 5924 , 2563758 , 15752	106517213 , 110957892 , 90122998 , 115023974 , 105812334 , 3903907 , 4418207 , 5539 , 812764 , 263207 , 2294627 , 253379
35	468295	42984759	42516464

36	5497129	19063716 , 45603681 , 33967036 , 17476337 , 55571430 , 47623191	13566587 , 40106552 , 28469907 , 11979208 , 50074301 , 42126062
37	127694	979939 , 339404	852245 , 211710
38	23137035	3050994 , 11066800 , 963947 , 18560400 , 28961300 , 125668949 , 243834886 , 238034886 , 237301442	20086041 , 12070235 , 22173088 , 4576635 , 5824265 , 102531914 , 220697851 , 214897851 , 214164407
41	2325086	4143 , 20040273 , 539806829	2320943 , 17715187 , 537481743
42	1450392	67574852 , 35130001 , 9700179 , 9980493 , 1902693 , 2921711 , 3059481 , 13234726 , 472552 , 2310002 , 6797122 , 70281100 , 98021078 , 19424683 , 42031750 , 53159810 , 80080287 , 40071324 , 40406267 , 3074044	66124460 , 33679609 , 8249787 , 8530101 , 452301 , 1471319 , 1609089 , 11784334 , 977840 , 859610 , 5346730 , 68830708 , 96570686 , 17974291 , 40581358 , 51709418 , 78629895 , 38620932 , 38955875 , 1623652
43	4751945	1120659 , 397774 , 506978 , 4972886 , 1999108 , 273301 , 482486 , 300642 , 525571 , 5232756 , 4408105 , 1300846 , 4152330 , 13272475 , 304994 , 8776784	3631286 , 4354171 , 4244967 , 220941 , 2752837 , 4478644 , 4269459 , 4451303 , 4226374 , 480811 , 343840 , 3451099 , 599615 , 8520530 , 4446951 , 4024839
44	676451	29668860 , 1022765555 , 1022765555 , 1022765555 , 1022765555 , 1022765555	28992409 , 1022089104 , 1022089104 , 1022089104 , 1022089104 , 1022089104
45	960952	7283146 , 4251675 , 11033258 , 2717611 , 4140745	6322194 , 3290723 , 10072306 , 1756659 , 3179793
46	4411	1322433 , 1322433 , 1322433 , 1322433	1318022 , 1318022 , 1318022 , 1318022
48	1285358	82227988 , 36904297	80942630 , 35618939

Table F.1: Mahalanobis distance of testing data for users with masqueraders using the SEA dataset. The masquerade distance column indicates how close the masquerade blocks are from the average training data point and the masquerader distance from avg column indicates how close the masquerade blocks are from the rest of the test blocks (ie. the average test block).

User	Average Test Distance	Masquerader Distance	Masquerader Distance from Avg
1	1062235	411405 , 411127 , 410939 , 102816 , 212953 , 102830 , 212644 , 173 , 103768 , 1643272 , 134 , 324401 , 2832587 , 1767676 , 210843 , 325147 , 3573257 , 1767610 , 3832555 , 1197566 , 410960 , 410874 , 1643281 , 410838 , 102717 , 924336 , 76 , 3697332 , 5032487 , 5032487	650830 , 651108 , 651296 , 959419 , 849282 , 959405 , 849591 , 1062062 , 958467 , 581037 , 1062101 , 737834 , 1770352 , 705441 , 851392 , 737088 , 2511022 , 705375 , 2770320 , 135331 , 651275 , 651361 , 581046 , 651397 , 959518 , 137899 , 1062159 , 2635097 , 3970252 , 3970252
2	638255	103400 , 1653373 , 58 , 413400 , 88 , 11 , 103496 , 530062 , 68 , 169 , 43 , 44 , 2583359 , 930058 , 2583371 , 103412 , 413410 , 41 , 413366 , 413535 , 413361 , 103539 , 103460 , 930019 , 103396 , 413375 , 1653351 , 26 , 930010 , 234	534855 , 1015118 , 638197 , 224855 , 638167 , 638244 , 534759 , 108193 , 638187 , 638086 , 638212 , 638211 , 1945104 , 291803 , 1945116 , 534843 , 224845 , 638214 , 224889 , 224720 , 224894 , 534716 , 534795 , 291764 , 534859 , 224880 , 1015096 , 638229 , 291755 , 638021
3	499824	15 , 1382396 , 929759 , 49 , 413261 , 47 , 124 , 103 , 106 , 2582310 , 103334 , 1652687 , 413322 , 103306 , 103307 , 1652668 , 103300 , 106250 , 2582310 , 2582310 , 2582310 , 2582302 , 2582297 , 1652678 , 2582293 , 413210 , 54 , 1652696 , 413191 , 103319	499809 , 882572 , 429935 , 499775 , 86563 , 499777 , 499700 , 499721 , 499718 , 2082486 , 396490 , 1152863 , 86502 , 396518 , 396517 , 1152844 , 396524 , 393574 , 2082486 , 2082486 , 2082486 , 2082478 , 2082473 , 1152854 , 2082469 , 86614 , 499770 , 1152872 , 86633 , 396505
4	558724	2580770 , 929055 , 103236 , 26 , 3716148 , 3716146 , 103248 , 412911 , 264 , 103232 , 3716147 , 412938 , 529042 , 529057 , 103238 , 943 , 27 , 32 , 212950 , 1651629 , 103234 , 103263 , 212918 , 8 , 929075 , 412913 , 103231 , 25 , 103291 , 103239	2022046 , 370331 , 455488 , 558698 , 3157424 , 3157422 , 455476 , 145813 , 558460 , 455492 , 3157423 , 145786 , 29682 , 29667 , 455486 , 557781 , 558697 , 558692 , 345774 , 1092905 , 455490 , 455461 , 345806 , 558716 , 370351 , 145811 , 455493 , 558699 , 455433 , 455485

5	1673663	321 , 527296 , 3709618 , 1648562 , 927344 , 927648 , 3709224 , 927473 , 5094037 , 103168 , 103168 , 412177 , 2575797 , 1648517 , 103095 , 103094 , 2575795 , 1648544 , 927293 , 3709148 , 927301 , 223 , 527336 , 5048609 , 1648980 , 1648533 , 1648709 , 103189 , 22 , 103060	1673342 , 1146367 , 2035955 , 25101 , 746319 , 746015 , 2035561 , 746190 , 3420374 , 1570495 , 1570495 , 1261486 , 902134 , 25146 , 1570568 , 1570569 , 902132 , 25119 , 746370 , 2035485 , 746362 , 1673440 , 1146327 , 3374946 , 24683 , 25130 , 24954 , 1570474 , 1673641 , 1570603
6	366014	100120 , 100117 , 12 , 10009585 , 2502439 , 801563 , 200408 , 900892 , 200518 , 100064 , 900906 , 14 , 400409 , 100109 , 400406 , 100049 , 182 , 100044 , 85 , 446 , 100046 , 900955 , 400400 , 100125 , 19 , 23 , 16 , 9585 , 9585 , 9585	265894 , 265897 , 366002 , 9643571 , 2136425 , 435549 , 165606 , 534878 , 165496 , 265950 , 534892 , 366000 , 34395 , 265905 , 34392 , 265965 , 365832 , 265970 , 365929 , 365568 , 265968 , 534941 , 34386 , 265889 , 365995 , 365991 , 365998 , 356429 , 356429 , 356429
7	485826	400655 , 400609 , 100179 , 401867 , 401867 , 103827 , 83 , 100175 , 100216 , 34 , 100120 , 100154 , 70 , 3606492 , 901695 , 49 , 43 , 405096 , 400790 , 201464 , 50 , 272 , 100302 , 1602915 , 901758 , 100207 , 901649 , 44 , 100179 , 400552	85171 , 85217 , 385647 , 83959 , 83959 , 381999 , 485743 , 385651 , 385610 , 485792 , 385706 , 385672 , 485756 , 3120666 , 415869 , 485777 , 485783 , 80730 , 85036 , 284362 , 485776 , 485554 , 385524 , 1117089 , 415932 , 385619 , 415823 , 485782 , 385647 , 85274
8	726098	100401 , 52 , 100338 , 32 , 273 , 2610295 , 2506885 , 902823 , 3610442 , 902577 , 401191 , 1604666 , 53 , 129 , 101048 , 100307 , 31 , 100328 , 100735 , 401223 , 67 , 100368 , 100311 , 6418234 , 8122954 , 1604565 , 401166 , 33 , 526 , 100343	625697 , 726046 , 625760 , 726066 , 725825 , 1884197 , 1780787 , 176725 , 2884344 , 176479 , 324907 , 878568 , 726045 , 725969 , 625050 , 625791 , 726067 , 625770 , 625363 , 324875 , 726031 , 625730 , 625787 , 5692136 , 7396856 , 878467 , 324932 , 726065 , 725572 , 625755

9	657270	102949 , 102956 , 411817 , 2573991 , 102983 , 411854 , 1905910 , 103163 , 411783 , 3844152 , 3844150 , 1905977 , 103103 , 103056 , 411851 , 1647072 , 2705931 , 103041 , 214318 , 926493 , 926613 , 3844168 , 2574342 , 411948 , 39 , 103052 , 104 , 3705986 , 411834 , 1647082	554321 , 554314 , 245453 , 1916721 , 554287 , 245416 , 1248640 , 554107 , 245487 , 3186882 , 3186880 , 1248707 , 554167 , 554214 , 245419 , 989802 , 2048661 , 554229 , 442952 , 269223 , 269343 , 3186898 , 1917072 , 245322 , 657231 , 554218 , 657166 , 3048716 , 245436 , 989812
10	585075	103024 , 4180 , 83 , 926506 , 411801 , 38 , 49 , 21 , 103062 , 3844134 , 36 , 92 , 40 , 40 , 117 , 211820 , 2105894 , 3844131 , 1373556 , 926541 , 411829 , 103048 , 102953 , 14 , 2573542 , 102988 , 926498 , 30 , 103106 , 411822	482051 , 580895 , 584992 , 341431 , 173274 , 585037 , 585026 , 585054 , 482013 , 3259059 , 585039 , 584983 , 585035 , 585035 , 584958 , 373255 , 1520819 , 3259056 , 788481 , 341466 , 173246 , 482027 , 482122 , 585061 , 1988467 , 482087 , 341423 , 585045 , 481969 , 173253
11	869246	1648527 , 103046 , 1648607 , 412224 , 103086 , 1648498 , 103053 , 103038 , 927283 , 927329 , 92 , 927299 , 8345510 , 1648496 , 927285 , 23 , 412177 , 33 , 103048 , 16 , 12 , 103369 , 412139 , 412145 , 2575776 , 5345509 , 927365 , 927287 , 412142 , 103057	779281 , 766200 , 779361 , 457022 , 766160 , 779252 , 766193 , 766208 , 58037 , 58083 , 869154 , 58053 , 7476264 , 779250 , 58039 , 869223 , 457069 , 869213 , 766198 , 869230 , 869234 , 765877 , 457107 , 457101 , 1706530 , 4476263 , 58119 , 58041 , 457104 , 766189
12	253575	102967 , 411872 , 526651 , 136 , 103244 , 27 , 102974 , 81 , 102983 , 70 , 46 , 39 , 149 , 121 , 180 , 102989 , 10 , 42 , 103308 , 102985 , 103036 , 14 , 12 , 926630 , 1647242 , 79 , 926576 , 411989 , 411950 , 411847	150608 , 158297 , 273076 , 253439 , 150331 , 253548 , 150601 , 253494 , 150592 , 253505 , 253529 , 253536 , 253426 , 253454 , 253395 , 150586 , 253565 , 253533 , 150267 , 150590 , 150539 , 253561 , 253563 , 673055 , 1393667 , 253496 , 673001 , 158414 , 158375 , 158272

13	4460713	103542 , 413363 , 930094 , 213397 , 213413 , 103621 , 8370055 , 10333406 , 8370055 , 10333406 , 6613378 , 5063373 , 413355 , 8370069 , 10333406 , 10333406 , 10333406 , 10333406 , 10333406 , 10333406 , 413661 , 1383645 , 1653647 , 930041 , 3720108 , 2583425 , 3720041 , 333406 , 333406 , 333406	4357171 , 4047350 , 3530619 , 4247316 , 4247300 , 4357092 , 3909342 , 5872693 , 3909342 , 5872693 , 2152665 , 602660 , 4047358 , 3909356 , 5872693 , 5872693 , 5872693 , 5872693 , 5872693 , 5872693 , 4047052 , 3077068 , 2807066 , 3530672 , 740605 , 1877288 , 740672 , 4127307 , 4127307 , 4127307
14	250691	413305 , 2575983 , 3594003 , 927388 , 103162 , 9 , 27 , 9 , 412141 , 103092 , 757 , 1169 , 103080 , 23 , 98 , 16 , 38 , 99 , 567 , 89 , 103060 , 29 , 24 , 71 , 120 , 927308 , 412163 , 26 , 24 , 85	162614 , 2325292 , 3343312 , 676697 , 147529 , 250682 , 250664 , 250682 , 161450 , 147599 , 249934 , 249522 , 147611 , 250668 , 250593 , 250675 , 250653 , 250592 , 250124 , 250602 , 147631 , 250662 , 250667 , 250620 , 250571 , 676617 , 161472 , 250665 , 250667 , 250606
15	506681	27 , 21 , 103638 , 414498 , 1058014 , 1657983 , 18 , 15 , 103635 , 103673 , 13 , 414498 , 24 , 31 , 103634 , 12 , 103641 , 20 , 1657984 , 3730452 , 103635 , 8 , 53 , 1530452 , 103636 , 932632 , 414514 , 414506 , 414546 , 19	506654 , 506660 , 403043 , 92183 , 551333 , 1151302 , 506663 , 506666 , 403046 , 403008 , 506668 , 92183 , 506657 , 506650 , 403047 , 506669 , 403040 , 506661 , 1151303 , 3223771 , 403046 , 506673 , 506628 , 1023771 , 403045 , 425951 , 92167 , 92175 , 92135 , 506662
16	1378955	47 , 5066809 , 414104 , 3722549 , 1654690 , 930689 , 103410 , 103413 , 103420 , 103535 , 413633 , 3722586 , 930643 , 103411 , 103415 , 1054580 , 2666837 , 2585111 , 930687 , 930761 , 103434 , 1654477 , 6617858 , 413629 , 103487 , 413738 , 930656 , 340400 , 340400 , 340400	1378908 , 3687854 , 964851 , 2343594 , 275735 , 448266 , 1275545 , 1275542 , 1275535 , 1275420 , 965322 , 2343631 , 448312 , 1275544 , 1275540 , 324375 , 1287882 , 1206156 , 448268 , 448194 , 1275521 , 275522 , 5238903 , 965326 , 1275468 , 965217 , 448299 , 1038555 , 1038555 , 1038555

17	302606	412931 , 60 , 85 , 2583975 , 103239 , 103283 , 31 , 23 , 103320 , 13 , 38 , 20 , 10 , 9 , 459 , 131 , 8 , 11 , 103545 , 80 , 103518 , 929788 , 25 , 413008 , 395 , 45 , 413443 , 96 , 470 , 145	110325 , 302546 , 302521 , 2281369 , 199367 , 199323 , 302575 , 302583 , 199286 , 302593 , 302568 , 302586 , 302596 , 302597 , 302147 , 302475 , 302598 , 302595 , 199061 , 302526 , 199088 , 627182 , 302581 , 110402 , 302211 , 302561 , 110837 , 302510 , 302136 , 302461
18	282492	533343 , 103818 , 533424 , 53 , 103714 , 103766 , 414825 , 94 , 414824 , 414844 , 414846 , 103712 , 414941 , 103726 , 20 , 91 , 103748 , 41 , 79 , 100 , 94 , 414846 , 933433 , 414908 , 76 , 103724 , 103735 , 370417 , 370417 , 370417	250851 , 178674 , 250932 , 282439 , 178778 , 178726 , 132333 , 282398 , 132332 , 132352 , 132354 , 178780 , 132449 , 178766 , 282472 , 282401 , 178744 , 282451 , 282413 , 282392 , 282398 , 132354 , 650941 , 132416 , 282416 , 178768 , 178757 , 87925 , 87925 , 87925
19	724544	1653741 , 1653721 , 1653717 , 413484 , 413525 , 413482 , 103529 , 413439 , 103393 , 413535 , 507 , 103593 , 930331 , 444 , 288 , 213 , 103403 , 103385 , 103823 , 1654774 , 10335612 , 413461 , 414220 , 1653869 , 3721065 , 6614886 , 2583936 , 2583909 , 2583906 , 3720856	929197 , 929177 , 929173 , 311060 , 311019 , 311062 , 621015 , 311105 , 621151 , 311009 , 724037 , 620951 , 205787 , 724100 , 724256 , 724331 , 621141 , 621159 , 620721 , 930230 , 9611068 , 311083 , 310324 , 929325 , 2996521 , 5890342 , 1859392 , 1859365 , 1859362 , 2996312
20	1169480	103049 , 1648807 , 6594009 , 103047 , 33 , 412205 , 176 , 103299 , 103164 , 412544 , 19 , 103084 , 412173 , 103126 , 43 , 103050 , 103203 , 3709115 , 412143 , 13 , 103058 , 103094 , 2575781 , 412130 , 2575784 , 1648710 , 1648686 , 103177 , 412647 , 46	1066431 , 479327 , 5424529 , 1066433 , 1169447 , 757275 , 1169304 , 1066181 , 1066316 , 756936 , 1169461 , 1066396 , 757307 , 1066354 , 1169437 , 1066430 , 1066277 , 2539635 , 757337 , 1169467 , 1066422 , 1066386 , 1406301 , 757350 , 1406304 , 479230 , 479206 , 1066303 , 756833 , 1169434

21	182181	24 , 37 , 102809 , 149 , 411132 , 58 , 43 , 20 , 102935 , 102822 , 44 , 44 , 925229 , 2569506 , 102841 , 102827 , 69 , 71 , 39 , 46 , 158 , 84 , 102819 , 52 , 49 , 86 , 24 , 64 , 32 , 56	182157 , 182144 , 79372 , 182032 , 228951 , 182123 , 182138 , 182161 , 79246 , 79359 , 182137 , 182137 , 743048 , 2387325 , 79340 , 79354 , 182112 , 182110 , 182142 , 182135 , 182023 , 182097 , 79362 , 182129 , 182132 , 182095 , 182157 , 182117 , 182149 , 182125
22	259007	70 , 454 , 411824 , 6588284 , 610 , 358 , 103106 , 927652 , 411946 , 926521 , 88 , 89 , 2573676 , 647129 , 103052 , 926488 , 411772 , 411772 , 411820 , 45 , 108 , 102972 , 102953 , 1647111 , 926600 , 102966 , 102959 , 103027 , 102987 , 102980	258937 , 258553 , 152817 , 6329277 , 258397 , 258649 , 155901 , 668645 , 152939 , 667514 , 258919 , 258918 , 2314669 , 388122 , 155955 , 667481 , 152765 , 152765 , 152813 , 258962 , 258899 , 156035 , 156054 , 1388104 , 667593 , 156041 , 156048 , 155980 , 156020 , 156027
23	2412947	81 , 3697648 , 411488 , 91 , 924412 , 5032809 , 44 , 12 , 1643408 , 411200 , 924464 , 2567796 , 102843 , 2632812 , 2597671 , 5032834 , 6573472 , 5032798 , 102735 , 102746 , 410951 , 3373462 , 5032862 , 80 , 102927 , 410971 , 139928 , 2402 , 104669 , 924441	2412866 , 1284701 , 2001459 , 2412856 , 1488535 , 2619862 , 2412903 , 2412935 , 769539 , 2001747 , 1488483 , 154849 , 2310104 , 219865 , 184724 , 2619887 , 4160525 , 2619851 , 2310212 , 2310201 , 2001996 , 960515 , 2619915 , 2412867 , 2310020 , 2001976 , 2273019 , 2410545 , 2308278 , 1488506
24	737959	602 , 213777 , 1197 , 104746 , 59 , 413344 , 20 , 24 , 931117 , 103345 , 23 , 413364 , 23 , 103353 , 103353 , 413379 , 207 , 2633425 , 15 , 103340 , 10 , 103410 , 413397 , 930084 , 6613442 , 104878 , 464305 , 153365 , 11 , 103349	737357 , 524182 , 736762 , 633213 , 737900 , 324615 , 737939 , 737935 , 193158 , 634614 , 737936 , 324595 , 737936 , 634606 , 634606 , 324580 , 737752 , 1895466 , 737944 , 634619 , 737949 , 634549 , 324562 , 192125 , 5875483 , 633081 , 273654 , 584594 , 737948 , 634610

25	1078703	3702906 , 925742 , 103080 , 102979 , 2571550 , 1645746 , 411849 , 6582916 , 6582916 , 1645740 , 411461 , 2571490 , 2571476 , 2571471 , 1645849 , 6582964 , 1645788 , 925889 , 925743 , 103224 , 1646202 , 1771669 , 103165 , 411977 , 24 , 103021 , 102870 , 32 , 102930 , 102914	2624203 , 152961 , 975623 , 975724 , 1492847 , 567043 , 666854 , 5504213 , 5504213 , 567037 , 667242 , 1492787 , 1492773 , 1492768 , 567146 , 5504261 , 567085 , 152814 , 152960 , 975479 , 567499 , 692966 , 975538 , 666726 , 1078679 , 975682 , 975833 , 1078671 , 975773 , 975789
26	700538	3700150 , 925200 , 925214 , 3700027 , 289 , 411472 , 27 , 26 , 54 , 411126 , 2569661 , 1387 , 102829 , 1644466 , 102805 , 1644485 , 102794 , 1644483 , 42 , 102834 , 63 , 241 , 411141 , 29 , 1644489 , 411203 , 102825 , 5036165 , 1644468 , 250	2999612 , 224662 , 224676 , 2999489 , 700249 , 289066 , 700511 , 700512 , 700484 , 289412 , 1869123 , 699151 , 597709 , 943928 , 597733 , 943947 , 597744 , 943945 , 700496 , 597704 , 700475 , 700297 , 289397 , 700509 , 943951 , 289335 , 597713 , 4335627 , 943930 , 700288
27	1022422	103385 , 929927 , 120127 , 24 , 91 , 28 , 103444 , 929958 , 1053190 , 1783156 , 1783133 , 103523 , 531587 , 616342 , 2109632 , 1783156 , 3777871 , 3862892 , 1183139 , 213320 , 1783157 , 1053271 , 1053210 , 929987 , 11 , 22 , 14 , 14 , 19 , 1383168	919037 , 92495 , 902295 , 1022398 , 1022331 , 1022394 , 918978 , 92464 , 30768 , 760734 , 760711 , 918899 , 490835 , 406080 , 1087210 , 760734 , 2755449 , 2840470 , 160717 , 809102 , 760735 , 30849 , 30788 , 92435 , 1022411 , 1022400 , 1022408 , 1022408 , 1022403 , 360746
28	517242	412950 , 1651671 , 104116 , 103305 , 929127 , 1651659 , 412976 , 2580711 , 929063 , 413018 , 413010 , 103289 , 412933 , 103280 , 31 , 103280 , 413084 , 35 , 412954 , 1651961 , 2581561 , 103293 , 198 , 103640 , 149 , 21 , 16 , 412944 , 103242 , 1651671	104292 , 1134429 , 413126 , 413937 , 411885 , 1134417 , 104266 , 2063469 , 411821 , 104224 , 104232 , 413953 , 104309 , 413962 , 517211 , 413962 , 104158 , 517207 , 104288 , 1134719 , 2064319 , 413949 , 517044 , 413602 , 517093 , 517221 , 517226 , 104298 , 414000 , 1134429

29	833433	152922 , 925738 , 3702893 , 1771486 , 1645770 , 525731 , 1645738 , 102878 , 411561 , 925736 , 925747 , 1645738 , 211468 , 525748 , 211461 , 411443 , 525756 , 925763 , 325747 , 102882 , 1171450 , 102891 , 411500 , 925735 , 925777 , 1045775 , 1045769 , 285762 , 285762 , 285762	680511 , 92305 , 2869460 , 938053 , 812337 , 307702 , 812305 , 730555 , 421872 , 92303 , 92314 , 812305 , 621965 , 307685 , 621972 , 421990 , 307677 , 92330 , 507686 , 730551 , 338017 , 730542 , 421933 , 92302 , 92344 , 212342 , 212336 , 547671 , 547671 , 547671
30	545020	102870 , 102921 , 1645738 , 925739 , 467 , 53 , 411495 , 102951 , 102972 , 1645757 , 925788 , 103317 , 411438 , 411440 , 411443 , 926229 , 103040 , 103254 , 411462 , 1045744 , 411467 , 411444 , 411476 , 411484 , 102912 , 411457 , 102904 , 411451 , 411454 , 2571454	442150 , 442099 , 1100718 , 380719 , 544553 , 544967 , 133525 , 442069 , 442048 , 1100737 , 380768 , 441703 , 133582 , 133580 , 133577 , 381209 , 441980 , 441766 , 133558 , 500724 , 133553 , 133576 , 133544 , 133536 , 442108 , 133563 , 442116 , 133569 , 133566 , 2026434
31	1295081	107842 , 411777 , 412879 , 412873 , 442 , 526075 , 1646343 , 212418 , 3704186 , 349095 , 945943 , 1646310 , 1646366 , 411595 , 2572351 , 926058 , 411609 , 102936 , 411656 , 926083 , 102930 , 411587 , 3704188 , 2572355 , 3704373 , 926102 , 1646307 , 411618 , 112 , 411612	1187239 , 883304 , 882202 , 882208 , 1294639 , 769006 , 351262 , 1082663 , 2409105 , 945986 , 349138 , 351229 , 351285 , 883486 , 1277270 , 369023 , 883472 , 1192145 , 883425 , 368998 , 1192151 , 883494 , 2409107 , 1277274 , 2409292 , 368979 , 351226 , 883463 , 1294969 , 883469
32	336735	328625 , 177 , 103161 , 21 , 16 , 13 , 40 , 103193 , 114 , 458 , 413734 , 1779075 , 213031 , 212755 , 928417 , 412562 , 103155 , 928260 , 103154 , 2186 , 528472 , 245 , 80 , 81 , 32 , 103226 , 2126 , 12401 , 528293 , 88	8110 , 336558 , 233574 , 336714 , 336719 , 336722 , 336695 , 233542 , 336621 , 336277 , 76999 , 1442340 , 123704 , 123980 , 591682 , 75827 , 233580 , 591525 , 233581 , 334549 , 191737 , 336490 , 336655 , 336654 , 336703 , 233509 , 334609 , 324334 , 191558 , 336647

33	750113	24 , 412521 , 928154 , 2578166 , 1650052 , 412529 , 412591 , 102 , 103261 , 412533 , 1650026 , 103144 , 412514 , 1178149 , 2112520 , 412536 , 928162 , 412536 , 928258 , 928221 , 2578150 , 103156 , 103146 , 31 , 103160 , 3712529 , 1778145 , 103164 , 103140 , 412548	750089 , 337592 , 178041 , 1828053 , 899939 , 337584 , 337522 , 750011 , 646852 , 337580 , 899913 , 646969 , 337599 , 428036 , 1362407 , 337577 , 178049 , 337577 , 178145 , 178108 , 1828037 , 646957 , 646967 , 750082 , 646953 , 2962416 , 1028032 , 646949 , 646973 , 337565
34	1005626	103418 , 103389 , 930135 , 413345 , 930013 , 1653453 , 60 , 930020 , 1653442 , 21 , 103355 , 34 , 413420 , 413917 , 413341 , 530140 , 1653825 , 413485 , 930347 , 2583453 , 413700 , 64 , 930012 , 103351 , 103367 , 213498 , 213392 , 530011 , 1653437 , 530030	902208 , 902237 , 75491 , 592281 , 75613 , 647827 , 1005566 , 75606 , 647816 , 1005605 , 902271 , 1005592 , 592206 , 591709 , 592285 , 475486 , 648199 , 592141 , 75279 , 1577827 , 591926 , 1005562 , 75614 , 902275 , 902259 , 792128 , 792234 , 475615 , 647811 , 475596
35	1243743	104015 , 416014 , 416017 , 416013 , 416138 , 1400028 , 104005 , 20 , 104014 , 416052 , 28 , 104011 , 14 , 416029 , 9 , 416034 , 28 , 416033 , 104139 , 8 , 416098 , 2600019 , 936015 , 6656038 , 2744034 , 416085 , 104033 , 400052 , 400052 , 400052	1139728 , 827729 , 827726 , 827730 , 827605 , 156285 , 1139738 , 1243723 , 1139729 , 827691 , 1243715 , 1139732 , 1243729 , 827714 , 1243734 , 827709 , 1243715 , 827710 , 1139604 , 1243735 , 827645 , 1356276 , 307728 , 5412295 , 1500291 , 827658 , 1139710 , 843691 , 843691 , 843691
36	194921	26 , 4 , 3 , 417474 , 8 , 2 , 104364 , 10 , 4 , 104363 , 3 , 104364 , 15 , 10 , 417427 , 17 , 14 , 4368 , 104363 , 24 , 417431 , 939206 , 6 , 8 , 12 , 6 , 6 , 435604 , 435604 , 435604	194895 , 194917 , 194918 , 222553 , 194913 , 194919 , 90557 , 194911 , 194917 , 90558 , 194918 , 90557 , 194906 , 194911 , 222506 , 194904 , 194907 , 190553 , 90558 , 194897 , 222510 , 744285 , 194915 , 194913 , 194909 , 194915 , 194915 , 240683 , 240683 , 240683

37	154909	29 , 4 , 12 , 5 , 23 , 104682 , 104682 , 13 , 19 , 4 , 10 , 9 , 15 , 17 , 104691 , 7 , 6 , 6 , 8 , 104690 , 56 , 104704 , 104711 , 218718 , 104714 , 6 , 9 , 467665 , 467665 , 467665	154880 , 154905 , 154897 , 154904 , 154886 , 50227 , 50227 , 154896 , 154890 , 154905 , 154899 , 154900 , 154894 , 154892 , 50218 , 154902 , 154903 , 154903 , 154901 , 50219 , 154853 , 50205 , 50198 , 63809 , 50195 , 154903 , 154900 , 312756 , 312756 , 312756
38	792071	5081558 , 1133490 , 10370467 , 50151 , 104142 , 13 , 9 , 9 , 9 , 25 , 414849 , 1659291 , 9 , 9 , 8 , 200093 , 200089 , 9 , 54 , 10 , 10 , 59 , 50108 , 46 , 85 , 25 , 25 , 5081533 , 3733370 , 1659375	4289487 , 341419 , 9578396 , 741920 , 687929 , 792058 , 792062 , 792062 , 792062 , 792046 , 377222 , 867220 , 792062 , 792062 , 792063 , 591978 , 591982 , 792062 , 792017 , 792061 , 792061 , 792012 , 741963 , 792025 , 791986 , 792046 , 792046 , 4289462 , 2941299 , 867304
39	607614	2 , 103707 , 103705 , 8570419 , 2733499 , 8400098 , 3099 , 3076 , 1976 , 136 , 103898 , 414851 , 933433 , 933422 , 3733363 , 488 , 3069 , 2 , 214817 , 103716 , 33 , 2 , 2 , 5 , 6 , 12 , 2 , 6800097 , 414987 , 2592710	607612 , 503907 , 503909 , 7962805 , 2125885 , 7792484 , 604515 , 604538 , 605638 , 607478 , 503716 , 192763 , 325819 , 325808 , 3125749 , 607126 , 604545 , 607612 , 392797 , 503898 , 607581 , 607612 , 607612 , 607609 , 607608 , 607602 , 607612 , 6192483 , 192627 , 1985096
40	605449	1975693 , 2500338 , 6 , 2 , 104141 , 104171 , 15 , 1 , 2 , 132 , 84 , 36 , 1 , 1360 , 9 , 382 , 1164 , 112666 , 15 , 1 , 2 , 99 , 121 , 11 , 4 , 66 , 10 , 138973 , 22 , 141565	1370244 , 1894889 , 605443 , 605447 , 501308 , 501278 , 605434 , 605448 , 605447 , 605317 , 605365 , 605413 , 605448 , 604089 , 605440 , 605067 , 604285 , 492783 , 605434 , 605448 , 605447 , 605350 , 605328 , 605438 , 605445 , 605383 , 605439 , 466476 , 605427 , 463884

41	356415	1986483 , 267352 , 866112 , 18 , 8 , 416532 , 537215 , 525 , 104149 , 2748749 , 2148771 , 416569 , 1666134 , 104172 , 26 , 53 , 104168 , 104138 , 416532 , 104141 , 416532 , 416553 , 104137 , 416548 , 70 , 416534 , 34 , 537218 , 33 , 537218	1630068 , 89063 , 509697 , 356397 , 356407 , 60117 , 180800 , 355890 , 252266 , 2392334 , 1792356 , 60154 , 1309719 , 252243 , 356389 , 356362 , 252247 , 252277 , 60117 , 252274 , 60117 , 60138 , 252278 , 60133 , 356345 , 60119 , 356381 , 180803 , 356382 , 180803

Table F.2: Mahalanobis distance of testing data for users with masqueraders using the Greenberg dataset. The masquerade distance column indicates how close the masquerade blocks are from the average training data point and the masquerader distance from avg column indicates how close the masquerade blocks are from the rest of the test blocks (ie. the average test block).