ON THE SUBSPACE LEARNING FOR NETWORK ATTACK DETECTION

THIAGO PEREIRA DE BRITO VIEIRA

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ON THE SUBSPACE LEARNING FOR NETWORK ATTACK DETECTION

THIAGO PEREIRA DE BRITO VIEIRA

ORIENTADOR: JOÃO PAULO C. LUSTOSA DA COSTA, PROF. DR.-ING.
COORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR, PROF. DR.

TESE DE DOUTORADO EM ENGENHARIA
ELÉTRICA

PUBLICAÇÃO: PPGEE.TD-147/2019

BRASÍLIA/DF: JULHO - 2019

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ON THE SUBSPACE LEARNING FOR NETWORK ATTACK DETECTION

THIAGO PEREIRA DE BRITO VIEIRA

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB
(PRESIDENTE DA COMISSÃO)

WILLIAM FERREIRA GIOZZA, Dr., ENE/UNB
(EXAMINADOR INTERNO)

JACOB SCHARCANSKI, Dr., UFRGS
(EXAMINADOR EXTERNOR)

ANDRÉ LIMA FERRER DE ALMEIDA, Dr., UFC
(EXAMINADOR EXTERNO)

Brasília, 12 de julho de 2019.

FICHA CATALOGRÁFICA

VIEIRA, THIAGO PEREIRA DE BRITO
On the Subspace Learning for Network Attack Detection
[Distrito Federal] 2019.
xi, 111p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2019).
Tese de doutorado – Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica
1. Network Attack Detection   2. Anomaly Detection
3. Subspace Learning   4. Robust Subspace Learning
I. ENE/FT/UnB   II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

VIEIRA, T. P. B. (2019). On the Subspace Learning for Network Attack Detection , Tese de doutorado, Publicação PPGEE.TD-147/2019, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 111p.

CESSÃO DE DIREITOS

AUTOR: Thiago Pereira de Brito Vieira
TÍTULO: On the Subspace Learning for Network Attack Detection .
GRAU: Doutor   ANO: 2019

Thiago P de B Vieira

Thiago Pereira de Brito Vieira

Departamento de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Eu dedico esta tese a João Emídio, meu pai, pela alegria que demonstrou ao me ver ingressar neste desafio e pela felicidade que teria tido ao ver os resultados obtidos.

Agradecimentos

Primeiramente eu gostaria de agradecer a Deus pela vida, saúde e todas oportunidades criadas em minha vida.

Agradeço aos meus pais, João e Ana, e ao meu irmão André, por todo o amor, carinho e incentivos para que eu possa sempre buscar crescimento pessoal e profissional, além de sempre me apoiarem nas minhas decisões e se mostrarem preocupados e empenhados em me ajudar a alcançar meus objetivos.

Agradeço à Alynne, minha esposa, por todo o amor e paciência durante todo nosso relacionamento, principalmente nestes intensos anos de doutorado, em que foram essenciais suas palavras de apoio nos momentos difíceis e sua descontração para me dar mais engergia e vontade de seguir em frente.

Agradeço ao meu filho Lucas, pela alegria e energia positiva que sempre me fortaleceu, mesmo nos momentos em que eu não pude estar ao seu lado, em virture da dedicação necessária aos estudos e produção científica.

Agradeço a todos que me orientaram e forneceram algum ensinamento durante o doutorado, em especial ao Prof. João Paulo Lustosa pelo acolhimento, apoio, orientações, cobranças e todos os importantes ensinamentos durante estes meses. Agradeço ao Prof. Rafael Timóteo por todos os ensinamentos e orientações em momentos importantes da minha pesquisa, pelos artigos que produzimos em conjunto, por aceitar ser meu coorientador e por presidir a banca da minha defesa de tese. Também agradeço ao Eduardo Kalil, pelo apoio, dúvidas e provocações que me ajudaram a visualizar melhorias importantes para o nosso trabalho.

Agradeço ao Prof. Jacob, ao Prof. André, ao Prof. Willian, ao Prof. Flávio e ao Ricardo por aceitarem fazer parte da banca da minha qualificação e de defesa de tese, como também pelas valiosas críticas construtivas e contribuições para a mehoria do meu trabalho.

Agradeço à Agência Nacional de Telecomunicações (Anatel), por permitir e proporcionar mais um aprendizado na minha vida. Gostaria de agradecer especialmente a Guilherme Chehab e Maria Lúcia Valadares por autorizarem o desafio de cursar um doutorado em concomitância com as atividades profissionais que desempenho. Agradeço a Eder Gualberto, pelo apoio ao longo destes anos desafiadores.

Agradeço a todos os amigos que fiz durante o doutorado, que contribuiram para que estes dias desafiadores fossem mais agradáveis. Finalmente, gostaria de agradecer a todos aqueles que colaboraram direta ou indiretamente na realização deste trabalho.

Muito Obrigado!!!

Resumo

O custo com todos os tipos de ciberataques tem crescido nas organizações. A casa branca do goveno norte americano estima que atividades cibernéticas maliciosas custaram em 2016 um valor entre US$57 bilhões e US$109 bilhões para a economia norte americana. Recentemente, é possível observar um crescimento no número de ataques de negação de serviço, botnets, invasões e ransomware.

A Accenture argumenta que 89% dos entrevistados em uma pesquisa acreditam que tecnologias como inteligência artificial, aprendizagem de máquina e análise baseada em comportamentos, são essenciais para a segurança das organizações. É possível adotar abordagens semi-supervisionada e não-supervisionadas para implementar análises baseadas em comportamentos, que podem ser aplicadas na detecção de anomalias em tráfego de rede, sem a ncessidade de dados de ataques para treinamento.

Esquemas de processamento de sinais têm sido aplicados na detecção de tráfegos maliciosos em redes de computadores, através de abordagens não-supervisionadas que mostram ganhos na detecção de ataques de rede e na detecção e anomalias.

A detecção de anomalias pode ser desafiadora em cenários de dados desbalanceados, que são casos com raras ocorrências de anomalias em comparação com o número de eventos normais. O desbalanceamento entre classes pode comprometer o desempenho de algoritmos traficionais de classificação, através de um viés para a classe predominante, motivando o desenvolvimento de algoritmos para detecção de anomalias em dados desbalanceados.

Alguns algoritmos amplamente utilizados na detecção de anomalias assumem que observações legítimas seguem uma distribuição Gaussiana. Entretanto, esta suposição pode não ser observada na análise de tráfego de rede, que tem suas variáveis usualmente caracterizadas por distribuições assimétricas ou de cauda pesada. Desta forma, algoritmos de detecção de anomalias têm atraído pesquisas para se tornarem mais discriminativos em distribuições assimétricas, como também para se tornarem mais robustos à corrupção e capazes de lidar com problemas causados pelo desbalanceamento de dados.

Como uma primeira contribuição, foi proposta a Autosimilaridade (*Eigensimilarity* em inglês), que é uma abordagem baseada em conceitos de processamento de sinais com o objetivo de detectar tráfego malicioso em redes de computadores. Foi avaliada a acurácia e o desempenho da abordagem proposta através de cenários simulados e dos dados do DARPA 1998. Os experimentos mostram que Autosimilaridade detecta os ataques synflood, fraggle e varredura de portas com precisão, com detalhes e de uma forma automática e cega, i.e. em uma abordagem não-supervisionada.

Considerando que a assimetria de distribuições de dados podem melhorar a detecção de anomalias em dados desbalanceados e assimétricos, como no caso de tráfego de rede, foi proposta a Análise Robusta de Componentes Principais baseada em Momentos (ARCP-m), que é uma abordagem baseada em distâncias entre observações contaminadas e momentos calculados a partir subespaços robustos aprendidos através da Análise Robusta de Componentes Principais (ARCP), com o objetivo de detectar anomalias em dados assimétricos e em tráfego de rede.

Foi avaliada a acurácia do ARCP-m para detecção de anomalias em dados simulados, com distribuições assimétricas e de cauda pesada, como também para os dados do CTU-13. Os experimentos comparam nossa proposta com algoritmos amplamente utilizados para detecção de anomalias e mostra que a distância entre estimativas robustas e observações contaminadas pode melhorar a detecção de anomalias em dados assimétricos e a detecção de ataques de rede.

Adicionalmente, foi proposta uma arquitetura e abordagem para avaliar uma prova de conceito da Autosimilaridade para a detecção de comportamentos maliciosos em aplicações móveis corporativas. Neste sentido, foram propostos cenários, variáveis e abordagem para a análise de ameaças, como também foi avaliado o tempo de processamento necessário para a execução do Autosimilaridade em dispositivos móveis.

Palavras-chave: Detecção de Anomalias, Detecção de Ataques de Rede, Dados Desbalanceados, Seleção de Ordem do Modelo (SOM), Similaridade de Autovetores, Análise Robusta de Componentes Principais (ARCP).

Abstract

The cost of all types of cyberattacks is increasing for global organizations. The Whitehouse of the U.S. government estimates that malicious cyber activity cost the U.S. economy between US$57 billion and US$109 billion in 2016. Recently, it is possible to observe an increasing in numbers of Denial of Service (DoS), botnets, malicious insider and ransomware attacks.

Accenture consulting argues that 89% of survey respondents believe breakthrough technologies, like artificial intelligence, machine learning and user behavior analytics, are essential for securing their organizations. To face adversarial models, novel network attacks and counter measures of attackers to avoid detection, it is possible to adopt unsupervised or semi-supervised approaches for network anomaly detection, by means of behavioral analysis, where known anomalies are not necessaries for training models.

Signal processing schemes have been applied to detect malicious traffic in computer networks through unsupervised approaches, showing advances in network traffic analysis, in network attack detection, and in network intrusion detection systems.

Anomalies can be hard to identify and separate from normal data due to the rare occurrences of anomalies in comparison to normal events. The imbalanced data can compromise the performance of most standard learning algorithms, creating bias or unfair weight to learn from the majority class and reducing detection capacity of anomalies that are characterized by the minority class. Therefore, anomaly detection algorithms have to be highly discriminating, robust to corruption and able to deal with the imbalanced data problem.

Some widely adopted algorithms for anomaly detection assume a Gaussian distributed data for legitimate observations, however this assumption may not be observed in network traffic, which is usually characterized by skewed and heavy-tailed distributions.

As a first important contribution, we propose the Eigensimilarity, which is an approach based on signal processing concepts applied to detection of malicious traffic in computer networks. We evaluate the accuracy and performance of the proposed framework applied to a simulated scenario and to the DARPA 1998 data set. The performed experiments show that synflood, fraggle and port scan attacks can be detected accurately by Eigensimilarity and with great detail, in an automatic and blind fashion, i.e. in an unsupervised approach.

Considering that the skewness improves anomaly detection in imbalanced and skewed data, such as network traffic, we propose the Moment-based Robust Principal Component Analysis (m-RPCA) for network attack detection. The m-RPCA is a framework based on distances between contaminated observations and moments computed from a robust subspace learned by Robust Principal Component Analysis (RPCA), in order to detect anomalies from skewed data and network traffic. We evaluate the accuracy of the m-RPCA for anomaly detection on simulated data sets, with skewed and heavy-tailed distributions, and for the CTU-13 data set. The Experimental evaluation compares our proposal to widely adopted algorithms for anomaly detection and shows that the distance between robust estimates and contaminated observations can improve the anomaly detection on skewed data and the network attack detection.

Moreover, we propose an architecture and approach to evaluate a proof of concept of Eigensimilarity for malicious behavior detection on mobile applications, in order to detect possible threats in offline corporate mobile client. We propose scenarios, features and approaches for threat analysis by means of Eigensimilarity, and evaluate the processing time required for Eigensimilarity execution in mobile devices.

Keywords: Anomaly Detection, Network Attack Detection, Imbalanced Data, Principal Component Analysis (PCA), Eigenvector Similarity, Robust Principal Component Analysis (RPCA).

Contents

List of Figures

List of Tables

List of Acronyms

ADM Alternating Direction Method

ALM Augmented Lagrange Multiplier Method

APG Accelerated Proximal Gradient

BYOD Bring-Your-Own-Device

C&C Command and Control

CEP Complex Event Processing

CF Click Fraud

CSF Critical Success Factor

DDoS Distributed Denial of service

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DoS Denial of Service

EDA Exploratory Data Analysis

EDC Efficient Detection Criterion

EFT Exponential Fitting Test

EVD Eigenvalue Decomposition

FAM Fast Alternating Minimization

FF Fast Flux

GETV Greatest Eigenvalue Time Vector

GQM Goal-Question-Metric

ICMP Internet Control Message Protocol

IF Isolation Forest

IRC Internet Relay Chat

IRLS Iteratively Reweighted Least Squares

ISS Information Security System

IT Iterative Thresholding

HDLSS High-Dimension Low-Sample-Size

KNN k-Nearest Neighbors

LAC Log Analysis Center

LOF Local Outlier Factor

MCD Minimum Covariance Determinant

MD Mahalanobis Distance

MOS Model Order Selection

NIDS Network Intrusion Detection System

OCSVM One-Class Support Vector Machines

PCA Principal Component Analysis

PCP Principal Component Pursuit method

PS Port Scan

R2L Remote to Local

RDA Robust Deep Autoencoders

RFE Recursive Feature Elimination

RPCA Robust Principal Component Analysis

SVM Support Vector Machines

TCP Transmission Control Protocol

TCU Federal Court of Accounts

TIM Threat Intelligence Manager

U2R User to Root

UAL User Activity Logging

UDP User Datagram Protocol

1 Introduction

According to Gartner surveys and client feedback, security remains a top concern for business and IT leaders. Cybersecurity is also highly visible in the media because of privacy concerns, destructive attacks such as ransomware, fears around IoT hacking, and an increasingly visible effect of cybersecurity on geopolitics [7].

The Whitehouse of the U.S. government [8] estimates that malicious cyber activity cost the U.S. economy between US$57 billion and US$109 billion in 2016. According to Bissel *et al.* [9], the total annual cost of all types of cyberattacks is increasing. Malware and Web-based attacks continue to be the most expensive threats for global companies, corresponding to US$2.6 million and US$2.2 million, respectively. However, it is possible to observe an increasing of 10% of Denial of Service (DoS) attack, 12% of botnets, 15% of malicious insider and an increment of 21% in occurrences of ransomware, between 2017 and 2018 [9].

Accenture's study conducted by Bissel *et al.* [9] shows that the extent of the economic value may be at risk if security investments are not made wisely. Bissel *et al.* [9] shows that the size of opportunity varies by industry, with high tech subject to the greatest value at risk — US$753 billion — over the next five years, followed by US$ 642 billion for life sciences and US$505 billion for the automotive industry. Bissel *et al.* [9] analyze the total technology savings minus total technology spend in cybersecurity by organizations, reporting that security intelligence and threat sharing have been adopted by 67% of respondents and saving US$2.26 million. Bissel *et al.* [9] also reported that automation, artificial intelligence, and machine learning have been adopted by 38% of respondents and saving US$2.09 million, while the cyber and user behavior analytics have been adopted by 32% of respondents and saving US$1.72 million.

Additionally, Levi [10] highlights that Blizzard Entertainment was hit by a Distributed Denial of Service (DDoS) attack in 2017, when the downtime lasted an entire weekend, due to difficulties related to track the root cause of the anomaly. According to Levi [10], attacks like these can cost an enterprise more than US$2 million, which makes anomaly detection essential to protecting revenue.

Traditionally, cyber defense methods can be effective against well-known types of attacks, yet may fail against innovative malicious techniques [11]. In order to be able to detect and avoid network attacks and their variations, it is necessary to develop or improve techniques to achieve efficiency in resource consumption, processing capacity and response time. Moreover, it is crucial to obtain high detection accuracy and capacity to detect variations of malicious patterns. Several efforts and researches aim to avoid network attacks based on known attackers, fingerprints or behavioral analysis [12, 13].

An attacker can create a bot for malicious purposes to generate a DoS attack, to exploit or abuse a target application. A botnet is a collection of bots or systems for executing automated tasks, often in the form of compromised hosts, including desktops, mobile devices or things (as in IoT). Some recent common examples of botnets include Cyclone, Mirai, Nitol and Sentry MBA [14]. Distributed attacks organized by botnet has increased and demanded the development of counter measures, in order to detect and avoid unknown attacks or even to deal with adversarial changes of behavior, location and other patterns [15, 16].

According to Hevesi [17] in a Gartner's publication, Botnets scan the Internet looking for unprotected Internet of Things (IoT) devices — and there are projected to be 25 billion devices online in 2021. Mobile network providers are deploying 5G capabilities to their networks, and industry is looking for new ways to leverage the service to bring new capabilities to market. Hevesi argues that if just 1% was able to be added to a botnet and had access to 5G, the potential network throughput would be 250 million devices able to push 10 to 50 Gbps of network traffic in a flooding attack.

Cyber security systems can work in the following fashions: signature-based, anomaly-based or hybrid [18]. A signature or fingerprint can be seen as a sequence of data, behavior and rules which are often unique to known malware or attack types, allowing the identification of attackers that reproduce some signature, such as occurs in less sophisticated viruses and automated toolkits for security exploitation and intrusion. However, signature-based systems for attack detection have to deal with adversarial models and techniques that aims to avoid detection based on well-known patterns, such as instruction virtualization, packing, polymorphism, emulation, and metamorphism to write and change malicious codes that can evade the detection [19].

Anomaly-based systems for network attack detection focus on finding exceptional, suspicious or rare observations in network traffic that do not conform to the expected legitimate behavior [19]. In a general view, anomalies are referred to as outliers, novelties or deviations, and can be related to defects, diseases, network attacks, intrusion detection for cyber security, fraud detection for credit cards, and military surveillance issues. Additionally, anomaly detection techniques can be categorized by classification, statistical algorithms, information theory and cluster based algorithms, according to [20].

Hybrid systems for network attack detection exploit benefits of both signature-based and anomaly-based detection techniques, and attempt to detect known as well as unknown attacks [20].

Accenture consulting argues that 89% of survey respondents believe breakthrough technologies, like artificial intelligence, machine or deep learning, user behavior analytics, and blockchain, are essential for securing the future of their organizations [21]. To face the adversarial model, novel attacks and counter measures of attackers to avoid detection, it is possible to adopt unsupervised or semi-supervised approaches for network anomaly detection, by means of behavioral analysis, where known anomalies are not necessaries for training models [22].

Recently, signal processing schemes have been applied to detect malicious traffic in computer networks by means of unsupervised approaches, showing advances in network traffic analysis [23], in network attack detection [24], and in network anomaly detection based on network flows [25].

Furthermore, it is possible to observe that anomaly-based and behavioral-based solutions have been adopted for cyber security and attracting investments, as can be seen in the Featurespace case, which is a provider of adaptive behavioral analytics for fraud detection and risk management, that raised US$32.3 million from a funding round led by Insight Venture Partners and MissionOG [26].

Anomalies in the context of network traffic can be hard to identify and separate from legitimate data due to the rare occurrences of anomalies in comparison to legitimate events. Therefore, anomaly detection algorithms have to be highly discriminating, robust to corruption and able to deal with the imbalanced data problem [27]. Note that data corruption refers to outliers that can be part of the data, while the imbalanced data problem corresponds to data sets exhibiting significant imbalances of classes or rare events of some classes [28], which can be legitimate or malicious classes in network anomaly detection problems.

The imbalanced data can compromise the performance of most standard learning algorithms, creating bias or unfair weight to learn from the majority class and reducing detection capacity of anomalies that are characterized by the minority class. Hence, data analysis of imbalanced data is challenging for learning algorithms applied to classification problems of anomaly detection, novelty detection, fraud detection and network attack detection. However, anomaly-based algorithms, that rely on unsupervised or semi-supervised approaches, can be alternatives for network anomaly detection from imbalanced data [22].

Some widely adopted algorithms for anomaly detection assume a Gaussian distributed legitimate data [11], however this assumption may not be observed in real world problems, such as the case of network traffic analysis [29], where network traffic features are usually more characterized by skewed and heavy-tailed distributions [29, 30]. According to [31], the skeweness and heavy-tailed distributions can impact algorithms that rely on Gaussian distributed data, and can reveal characteristics that can be exploited in order to obtain accurate classifiers for network anomaly detection.

Therefore, this thesis focus on anomaly-based problems. More specifically, we focus on network attack detection and propose behavioral-based approaches for network anomaly detection through subspace learning techniques and similarity analysis.

Firstly, we propose the Eigensimilarity, which is an approach based on signal processing methods applied to detection of probe and flooding attacks in computer networks. We present an architecture and approach to evaluate a proof of concept of Eigensimilarity for malicious behavior detection on mobile applications, in order to detect possible threats in offline corporate mobile client. Additionally, we propose the Moment-based Robust Principal Component Analysis (m-RPCA), which is an approach based on distances between contaminated observations and moments computed from a robust subspace learned by Robust Principal Component Analysis (RPCA), in order to detect anomalies from imbalanced and skewed data, such as network traffic.

This chapter is organized as follows. The problem statement, the hypotheses formulation and the proposed approaches for network attack detection are introduced in Section 1.1. In Section 1.2 we present the main contributions of this thesis and in Section 1.3 we describe the thesis organization for the next chapters.

1.1 Problem Statement

Considering the previous described landscape, this thesis outlines the development and evaluation of approaches based on subspace learning for network attack detection, through methods to make the data discriminative and able to identify structures, hidden patterns and the most relevant information for anomaly detection. In Subsection 1.1.1 we present our hypothesis formulation and in Subsection 1.1.2 we describe the proposed approaches to answer the questions and validate the hypotheses.

1.1.1 Hypothesis Formulation

For the experimental evaluation of our proposals, we adopt a methodology based on aspects of Goal-Question-Metric (GQM) template [32] and define two questions to achieve our goal, which are:

- Q_1: Can the analysis of patterns from a learned subspace identify and detect anomalies in network traffic?
- Q_2: Can the robust subspace learning improve the anomaly detection in imbalanced and skewed data?

Our testing hypotheses are defined in Table 1.1, that describe the null hypotheses ($H_1^{(N)}$ and $H_2^{(N)}$) and alternative hypotheses ($H_1^{(A)}$ and $H_2^{(A)}$) for each previously defined question.

Table 1.1 Hypotheses to evaluate the defined questions

Alternative Hypothesis	**Null Hypothesis**	**Question**
$H_1^{(A)}$: A subspace learned by eigenvalue decomposition can be used to detect and identify network attacks.	$H_1^{(N)}$. A subspace learned by eigenvalue decomposition can not be used to detect and identify network attacks.	Q_1
$H_2^{(A)}$: An approach based on robust subspace learning improves the anomaly detection from imbalanced and skewed data.	$H_2^{(N)}$. An approach based on robust subspace learning does not improves the anomaly detection from imbalanced and skewed data.	Q_2

The hypotheses $H_1^{(A)}$ and $H_1^{(N)}$ are defined to evaluate if a subspace learned by eigenvalue decomposition are sensitive to outliers and can be used to detect network attacks. We define the hypotheses $H_2^{(A)}$ and $H_2^{(N)}$ to evaluate if the distance between contaminated data and robust moments learned by Robust Principal Component Analysis (RPCA) can improve the anomaly detection in simulated imbalanced and skewed data.

1.1.2 Proposals

In the context of anomaly-based schemes, this thesis proposes the Eigensimilarity, which is an approach based on subspace learning techniques for detection of malicious traffic in computer networks, by means of eigenvalue analysis, model order selection (MOS) and a similarity analysis between eigenvectors of estimated legitimate observations and observations of a time frame estimated as under attack.

In contrast to [33, 34, 35], MOS and eigenvalue analysis are applied to detect detailed time frames under attack. We evaluate the accuracy and performance of the proposed framework applied to an experimental scenario and to the DARPA 1998 data set [36], which is a well-known network traffic data set. Furthermore, this proposed approach is evaluated by a proof of concept regarding behavioral anomaly detection to detect possible threats to an offline corporate mobile app.

The skewness of anomalous and legitimate data can highlight features for improving anomaly detection in imbalanced data, and the distance between robust estimates of legitimate observations and contaminated data can be used for network attack detection. Therefore, we propose the Moment-based Robust Principal Component Analysis (m-RPCA), which is an approach based on distances of moments computed from a robust subspace learned by RPCA, for anomaly detection on imbalanced and skewed data. We evaluate the results of m-RPCA for anomaly detection on simulated imbalanced and skewed data, and evaluate the results of m-RPCA for network attack detection on CTU-13 data set.

Eigensimilarity and m-RPCA are frameworks for network attack detection by means of anomaly-based analysis from network traffic. In a simplified architecture for network attack detection shown by Figure 1.1, Eigensimilarity and m-RPCA are deployed as a module of an application firewall, which is responsible for secure and protect application communications, but can also work on network level.

An application firewall can implement behavioral analysis in order to detect attacks against application services, such as Spam or Click Fraud (CF), or can detect anomalies in network level, such as a DoS attack based on Ping flood.

The Figure 1.1 depicts the flow of network traffic between legitimate or malicious users to a corporate network. All income traffic shall be received and distributed by the load balancer, to be evaluated by an application firewall, which is is responsible for network attack detection. Therefore, the application firewall implements a high throughput traffic analyzer, to capture and parse the network traffic for further analysis, by means of Eigensimilarity and m-RPCA. The Figure 1.1 also depicts the use of Eigensimilarity as a

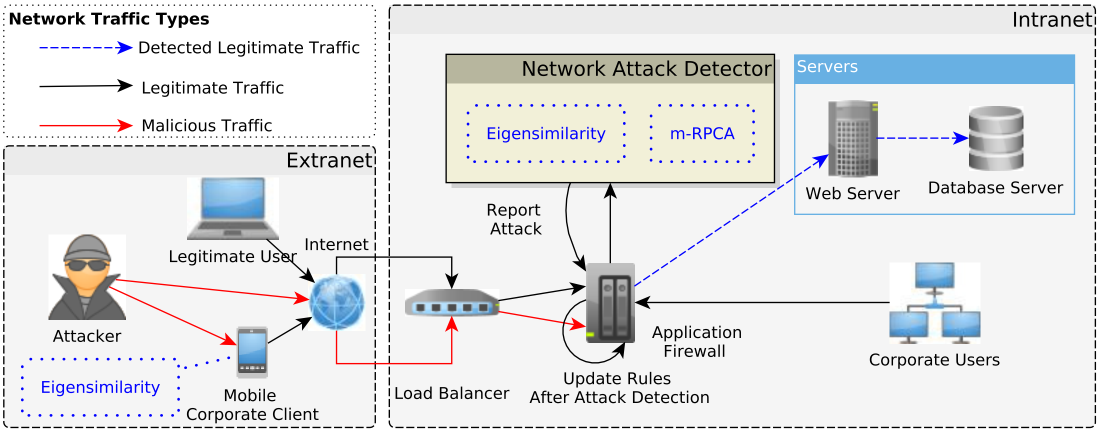


Figure 1.1 Simplified Architecture for Network Attack Detection.

module of an offline corporate mobile app, as a proof of concept regarding behavioral anomaly detection from user activities.

1.2 Contributions

We analyze problems related to detection of information security issues in network traffic and propose new approaches to improve malicious behavior detection through signal processing techniques based on subspace learning. The results of the work presented in this thesis provide the following publications and contributions:

1. T. P. B. Vieira, D. F. Tenório, J. P. C. da Costa, E. P. de Freitas, G. Del Galdo, and R. T.de Sousa Júnior, “Model order selection and eigen similarity based framework for detection and identification of network attacks, ” *Journal of Network and ComputerApplications*, vol. 90, pp. 26–41, 2017 [1].

   1.1. We propose the Eigensimilarity, which is an approach based on eigenvector similarity analysis for extracting detailed information about accurate time and network ports under network attack, and evaluate the accuracy and performance of the proposed framework applied to an experimental scenario and to the DARPA 1998 data set;

   1.2. We discuss the computational complexity of the Eigensimilarity and evaluate the required processing time for tested scenarios;

2. T. P. B. Vieira, J. P. C. L. da Costa, E. S. C. Vilaça, E. S. Gualberto, and R.

T.de Sousa Júnior, “Moment distances from robust subspace for network attack detection, ” *Journal of Network and Computer Applications*, To Appear [2].

2.1. We propose the m-RPCA, which is an approach based on distances of moments computed from a robust subspace learned by RPCA, for anomaly detection on imbalanced and skewed data, and evaluate the anomaly detection and network attack detection rates on simulated and real data sets.

3. T. Galibus, T. P. B. Vieira, E. P. de Freitas, R. d. O. Albuquerque, R. T.de Sousa Júnior, V. Krasnoproshin, A. Zaleski, H. Vissia, G. del Galdoet al., “Of-fline mode for corporate mobile client security architecture, ” *Mobile Networks and Applications*, pp. 1–17, 2017 [3].

   3.1. We propose an architecture and implement a proof of concept for offline behavioral analysis of a corporate mobile client, and discuss the processing time of the Eigensimilarity for mobile devices;

4. K. H. C. Ramos, R. T. de Sousa Junior, T. P. B. Vieira, and J. P. C. L. da Costa, “Discovering critical success factors for information technologies governance through bibliometric analysis of research publications in this domain, ” *International Information Institute (Tokyo). Information*, vol. 19, no. 6B, p. 2193, 2016. [4].

5. K. H. C. Ramos, T. P. B. Vieira, J. P. C. L. da Costa, and R. T. de Sousa Júnior, “Multidimensional analysis of critical success factors for it governance within the Brazilian federal public administration in the Light of External Auditing Data". *12th International CONTECSI*, 2015 [5].

   5.1. We propose a critical factors analysis based on Principal Component Analysis (PCA) for visual discriminant analysis, and presenting an approach based on Recursive Feature Elimination (RFE) combined with Support Vector Machine (SVM), in order to identify the Critical Success Factors (CSF) for IT governance.

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2, we propose the Eigensimilarity, which is an approach based on signal processing techniques for detection of malicious traffic in computer networks, based on eigenvalue analysis, model order selection (MOS) and similarity analysis. In Chapter 3 we present a proof of concept regarding the evaluation of an approach and architecture based on user behavior analysis through the Eigensimilarity [1], in order to detect threats in a mobile application. The m-RPCA is proposed in Chapter 4, where is presented the proposed approach based on distances of moments computed from a robust subspace learned by RPCA, for anomaly detection on imbalanced and skewed data. In Chapter 5 we draw the conclusions and the suggestions for future work. Furthermore, in Appendix A we present a critical factors analysis based on Principal Component Analysis (PCA), Recursive Feature Elimination (RFE) and Support Vector Machine (SVM), in order to identify the Critical Success Factors (CSF) for IT governance.

2 Eigensimilarity based Framework for Detection and Identification of Network Attacks

According to [17], at times, organizations relate their security measures to confidentiality and integrity, ignoring availability. However, according to [9], the global average annual cost of DoS attack was US$1.7 million in 2018.

The Denial of Service (DoS) attack attempts to deny access to system or network resources, attacking the availability of the service, by means of flooding techniques for consuming the available resources, or by means of subtle approaches which send small amount of data that can cause failure and unavailability of the system. In 2017 the top motivation behind DDoS attacks was criminals demonstrating attack capabilities, with gaming and criminal extortion attempts in second and third place, respectively [37].

With application layer Distributed Denial of Service (DDoS) attacks continuing to rise, vendors have begun adding DDoS mitigation features into solutions that protect web applications [17]. The leading solutions for monitoring and mitigation of DDoS attacks in application layer start with behavioral analytics and supplement with signature-based detection for known malicious application attacks [17].

Probe attacks aim to scan the networks to identify running services, open ports, running services and vulnerabilities that can be exploited. A probe attack is considered the first step in an attack to compromise a host or network. Although no specific damage is caused by these attacks, they are considered serious threats by [20] and [22]. Note that we refer to port scan as an attack, according to [20].

Anomaly-based systems for network attack detection focus on finding exceptional, suspicious or rare observations in network traffic that do not conform to the expected normal behavior [19]. It is possible to adopt unsupervised or semi-supervised approaches for network anomaly detection, by means of behavioral analysis, where known anomalies for training models are not necessary [22].

Recently, signal processing schemes have been applied for network anomaly detection in computer networks by means of unsupervised approaches, showing advances in network traffic analysis [23], in network attack detection [24], and in network anomaly detection based on network flows [25].

Considering that a subspace learned by eigenvalue decomposition can highlight anomalies for network attack detection by means of model order selection (MOS), and that the comparison between the principal eigenvectors can reveal anomalies when comparing legitimate and anomalous observations. In this chapter we propose the Eigensimilarity, which is an approach based on signal processing concepts applied to detection of malicious traffic in computer networks, with focus on DoS and portscan attacks.

The Eigensimilarity is based on eigenvalue analysis, model order selection (MOS) and similarity analysis between the principal eigenvectors. In contrast to [33, 34, 35], MOS and eigenvalue analysis are applied to detect time frames under attack, and the analysis of similarity between the principal eigenvectors are used for precise identification of the time and ports under attack. In addition, we evaluate the accuracy and performance of the proposed framework applied to an experimental scenario and to the DARPA 1998 data set [36], which is a well-known network traffic data set.

The performed experiments show that synflood, fraggle and port scan attacks can be accurately detected and with great detail in an automatic and blind fashion, i.e. in an unsupervised approach that does not require data for training, applying signal processing concepts for traffic modeling and through approaches based on MOS and principal eigenvector similarity analysis. The main contributions of the proposed framework are the capability to blindly detect time frames under network attack via MOS and eigen analysis, and the detailed identification of the network attack via principal eigenvector similarity analysis.

This chapter is organized as follows. In Section 2.1 the related works are discussed. We present the data model and the evaluated data sets in Section 2.2 and the proposed framework for blind and automatic detection of flood and probe attacks is described in Section 2.3. The discussion of the experimental validation and results are presented in Section 2.4, while the discussion of the computational complexity and evaluation of the required processing time for tested scenarios are presented in Section 2.5. In Section 2.6

we draw the conclusions and the suggestions for future work.

2.1 Related Works

Several methods have been proposed for the identification and characterization of malicious activity in computer networks. Classical methods typically employ data mining [38, 39, 36] and regular file analysis [40] to detect patterns that indicate the presence of specific attacks in network traffic.

Data mining is often used to describe the process of extracting useful information from large databases. Multiple methods of data mining are used in [38, 36] to analyze data flow in a network with the aim of identifying characteristics of malicious traffic in large scale environments. Researchers have applied data mining techniques in log analysis [39] to improve intrusion detection performance. However, data mining techniques used so far in network analysis require prior collection of large data sets, which is a limitation of several schemes for online analysis [38].

Regular file analysis [40] consists of traffic analysis for detecting known patterns that indicate the presence of attacks, applying statistical analysis to the study of collected traffic. An essential feature of this method is that it depends on prior knowledge of the details of the attacks to be identified, and also depends on previous log collection for traffic analysis and false positives reduction.

Principal Component Analysis (PCA) is a statistical technique commonly used for dimensionality reduction. It uses an orthogonal transformation to convert a set of correlated variables into a new subspace of linearly uncorrelated variables, by means of eigenvalue value decomposition, where the first principal components have the largest variance. PCA has been used in attack detection [41], considering that this technique is very sensitive to outliers and adopted for classification problems. However, PCA requires human intervention in order to identify abnormalities based on the eigenvalues profiles, if used without complementary techniques.

Callegari *et al.* [24] proposed a PCA-based method for identifying the traffic flows responsible for an anomaly detected at the aggregate level and evaluated their proposal through a data set with synthetic anomalies added in the data. However, Callegari *et al.* [24] focus on flood attack detection, not addressing probe attack detection, and their approach relies on visual analysis.

Lee *et al.* [42] presented the OverSampling PCA (osPCA), which allows one to determine the anomaly of the target instance according to the variation of the resulting dominant eigenvector obtained by similarity analysis and over sampling. In contrast to Lee *et al.* [42], the Eigensimilarity applies MOS for detection of time frames under attack and similarity analysis to extract details for detection of time and ports under attack. Additionally, Lee *et al.* [42] only evaluate their proposed scheme for covariance analysis, while we adopt an analysis based on sample covariance of zero mean variables and sample covariance of zero mean and unitary standard deviation variables, for flood and probe attacks, respectively.

Signal processing techniques have been successfully applied to network anomaly detection [25]. Lu and Ghorbani [25] proposed a network anomaly detection model based on network flow, wavelet approximation, and system identification theory. However, their work requires a training method to produce a prediction model for normal daily traffic and presents limitations on identification of behaviors without significant outliers, such as port scan attacks. Zonglin *et al.* [23] proposed a signal processing method to detect traffic anomaly with correlation analysis, where the correlation between traffic signals and the predicted traffic signals are used to reveal anomalies. Zonglin *et al.* [23] evaluated the correlation analysis for anomaly detection, but the work is not applied for probe attack detection and does not evaluate results for attack detection on DARPA data set.

The data collected from honeypot systems, such as captured traffic and operating system logs, can be analyzed to obtain information about attack techniques, general trends of threats and exploits [43]. Blind automatic detection of malicious traffic techniques have been developed for honeypots in [33, 34]. However, traffic on honeypot is simpler than real network traffic, because there are no running legitimate applications, i.e. background traffic, due to the fact that honeypots emulate behavior of a host within a network to deceive and lure attackers [44]. Since honeypots do not generate legitimate traffic, the amount of data captured in honeypots is significantly lower in comparison to a Network Intrusion Detection System (NIDS), which captures and analyzes the largest possible amount of network traffic [33].

MOS for blind identification of malicious activities in honeypots was proposed by us in [33], which evaluated criteria for selecting the model order, through simulations and comparing the order of the resulting model with the true model order.

The proposed framework does not require either a significant amount of logs to detect attacks, nor prior data collection, in order to make comparisons and evaluate the existence of malicious traffic. The proposed solution is automatic and blind for detection of time frames under probe and flood attacks through MOS and eigen analysis. Moreover, we apply principal eigenvector similarity analysis to identify details of time and ports under network attacks.

Ahmed *et al.* [20] highlight a observed lack of publicly available labeled data set for network anomaly detection, and discuss recent research related to overcome the issues of publicly available network intrusion evaluation data sets, including a list of current data sets and an evaluation of attack types and available labels. According to [20], the available data sets present limitations on availability of labeled traffic for classification of background, legitimate and malicious traffic. The data imbalance between legitimate and malicious traffic is a important concern for real world problems of network attack, but the principal data sets usually adopt a balanced distribution between legitimate and malicious traffic, what make the attack detection more feasible for general classifiers and machine learning algorithms than for anomaly detection algorithms. Moreover, according to Ahmed *et al.* [20], the most adopted data set are not up-to-date with current or novel attacks, tools, operational systems and network topology.

Several approaches for network attack detection uses the KDD 99 [45, 20, 36, 19] data sets for accuracy and performance evaluation, due to their availability and labeled attacks. Even though the KDD 99 data set are criticized by for the generation procedure and the risk of over-estimations of anomaly detection due to data redundancy, it still represents one of the few publicly available labeled data sets adopted by researchers [36, 19]. NSL-KDD [46] data set is the refined version of the KDD 99 data set that redundant data records are removed, in order to avoid biased classifications. Additionally, some approaches uses simulated [24] scenarios or non-public data sets their evaluations.

Since the proposed approach relies on a packet level analysis and the KDD 99 and NSL-KDD data sets adopt a traffic aggregation by connections, we consider the use of an experimental scenario on a real network and the DARPA 1998 data set, which is the source for the creation of the KDD 99 and NSL-KDD data sets. Note that the proposed approach is not based on learning or classification techniques, which are more susceptible to biased results caused by the issues in the DARPA/KDD data sets.

2.2 Data Model

In this work, scalars are denoted by italic letters $(a, b, A, B, \alpha, \beta)$, vectors by lowercase bold letters $(\boldsymbol{a}, \boldsymbol{b})$, matrices by uppercase bold letters $(\boldsymbol{A}, \boldsymbol{B})$, and $a_{i,j}$ denotes the (*i, j*) elements of the matrix $\boldsymbol{A}$. The superscripts $^{\mathrm{T}}$ and $^{-1}$ are used for matrix transposition and matrix inversion, respectively. The l_2 norm is denoted by $||\cdot||$. We define the operator $\mathrm{diag}(\cdot)$ that returns the vector of the main diagonal of a given matrix, the operator $\rightarrow$, which denotes the deletion of a given element from a set and the operator #, that returns the rank of a matrix, and the operator $\sim$ that sorts the elements of a vector in ascending order.

This section also presents details of the experimental scenario and the selected cases of the DARPA data set. In Subsection 2.2.1 we describe the environment and scenario adopted in order to reproduce flood and probe attacks. In Subsection 2.2.2 are presented how network traffic can be modeled as signal superposition. We detail, in Subsection 2.2.3, the traffic of synflood, fraggle and port scan attacks, and in Subsection 2.2.4 are discussed the use of the DARPA data set for evaluation of the proposed approach.

2.2.1 Analyzed Scenario and Data Collection

The environment of the analyzed scenario is composed by two computers and one router, with access to the Internet and to an internal network. In this scenario we performed the simulation of legitimate traffic, control, flood and port scan attacks. During the traffic generation, one computer assumes the role of the attacker, while the other is the victim, according to scenario represented by Figure 2.1. The victim generate legitimate traffic by means of ordinary use of Internet resources, and control traffic is constantly generated for purposes of network administration and monitoring.

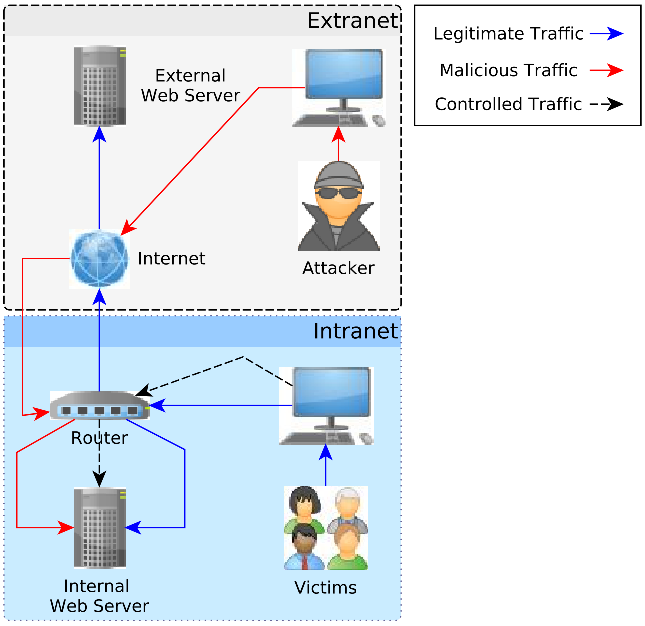


Figure 2.1 Scenario to reproduce legitimate traffic, flood and port scan.

In many organizations the web based traffic is predominant, since most of corporate services are web pages, customized web-based systems and cloud services. It is possible to characterize the traffic of a Dynamic Host Configuration Protocol (DHCP) service as an example of legitimate associated with the application layer, as well as it would be possible to classify seasonal and controlled traffic as legitimate traffic. For malicious traffic, three types of networks attacks are evaluated: synflood, fraggle and port scan. Here we refer to port scan as an attack, according to [20, 22], however it is one approach usually adopted for acquire information in order to perform an attack. These attacks are reproduced using well-known security tools, such as Nmap[1] for port scan, Metasploit[2] for synflood and Hping[3] to lead the fraggle attack.

A network traffic log is commonly formed by timestamp, protocol, source IP address, source port, destination IP address, destination port and additional information, according to the type of the used transport protocol. The following TCP traffic log is presented in order to exemplify the collected data:

```
21:00:34.099289 IP 192.168.1.102.34712 > 200.221.2.45.80: Flags
[S], seq 2424058224, win 14600, options [mss 1460, sackOK,TS
val 244136 ecr 0,nop,wscale 7], length 0
```

and the following to exemplify UDP traffic log:

```
21:24:42.484858 IP 192.168.1.102.68 > 192.168.1.1.67: BOOTP/DHCP,
Request from 00:26:9e:b7:82:be, length 300
```

In the proposed framework, the goal is to detect the anomalies only taking into account the traffic profile, i.e. specific information such as origin or destination IP, behavioral pattern or content of the attack are not considered. Therefore, IP spoofing or data encryption would not cause impact to the proposed approach and evaluation, since our proposal only relies on the timestamp (for sequencing) and destination port number.

[1] http://nmap.org

[2] http://www.metasploit.com

[3] http://hping.org

2.2.2 Modeling Data

By modeling the data set as a signal superposition, the network traffic ($\boldsymbol{X}$) can be characterized as a mixture of two components: legitimate traffic ($\boldsymbol{U}$) and malicious traffic ($\boldsymbol{N}$), according to the following expression:

$$\boldsymbol{X}^{(q)} = \boldsymbol{U}^{(q)} + \boldsymbol{N}^{(q)}, \tag{2.1}$$

where q represents the q-th time frame, which is a time aggregation of network traffic. The matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$ consists of M rows and N columns, where each row represents a communication port, and each column represents time bins of a defined size, such as one minute. Each element $x_{m,n}^{(q)}$ stands for the number packets that appears at n-th minute for the port m, during the q-th time frame.

The legitimate traffic $\boldsymbol{U}^{(q)}$ is characterized by user's ordinary operations and by legitimate traffic that are automatically generated for network management and for background services. The access to web pages by users or the name resolution by means of the Domain Name System (DNS) are examples of legitimate traffic generated by user operations, as can be seen in Figure 2.2, which depicts the legitimate traffic simulated to reproduce user's operations during the experiments.

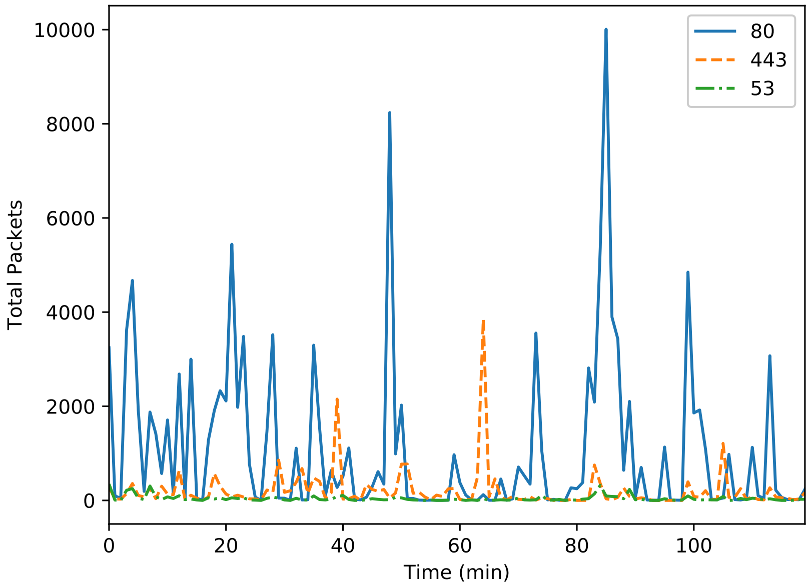


Figure 2.2 Traffic from user's operations.

The Figure 2.3 depicts an example of legitimate traffic of user independent operations, by means of traffic to ports 67 and 68, where it is possible to observe a low amount of packets. The acquisition of logical IP network address by means of DHCP is an example of legitimate traffic not associated with a user, where independently of any user operation, the machine receives an IP address, since it is configured to automatically perform a DHCP address request.

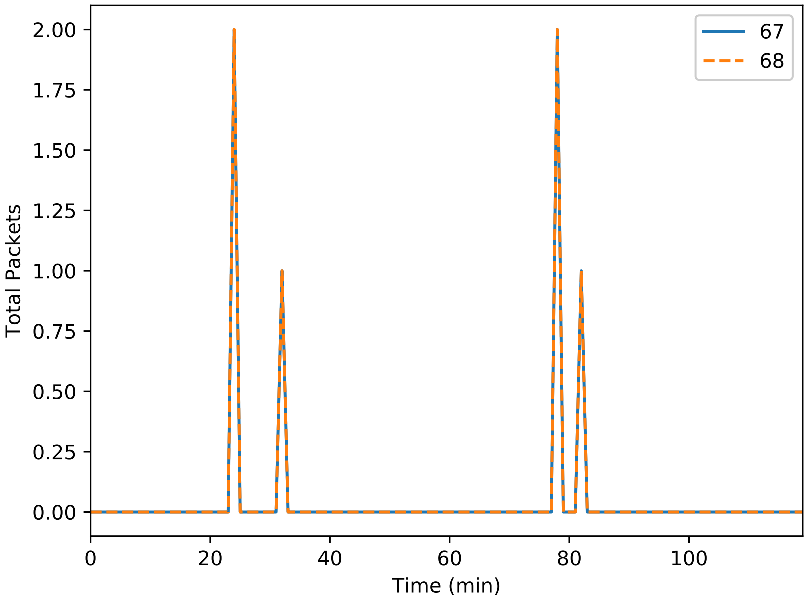


Figure 2.3 Network traffic of user independent operations for network management.

The traffic coming from a malicious activity, i.e. port scanning, synflood or fraggle attacks, is represented by the matrix $\boldsymbol{N}^{(q)}$. We define that if the rank of $\boldsymbol{X}^{(q)}$ is not zero, according to $\#\boldsymbol{X}^{(q)} \neq 0$, which denotes that the rank of the q-th $\boldsymbol{X}$ is different of 0, then there is malicious traffic in the evaluated time frame q. On the other hand, if the $\#\boldsymbol{X}^{(q)} = 0$, then there is no malicious traffic in time frame q. We show how to detect the $\#\boldsymbol{X}^{(q)}$, given only the matrix $\boldsymbol{X}^{(q)}$, in order to identify malicious traffic.

2.2.3 Synflood, Fraggle and Port scan

The network attacks evaluated by this work are: synflood, fraggle and port scan. The first two attacks can be qualified as flood or denial of service (DoS) attacks, while the last one can be qualified as probe or port scanning attack.

A DoS is an attempt by an attacker to prevent legitimate access to websites by overwhelming the amount of available bandwidth or resources of the computer system. DoS is implemented by either forcing targets to be unavailable through the exploiting of system vulnerabilities, or consuming resources through large amount of network traffic, characterizing flood attacks. Probe attacks scan computer and network systems to collect information about the host, such as open ports, topology, running software or version of technologies, in order to find vulnerabilities.

With respect to the synflood attacks, the attacker sends a large quantity and concurrent successive SYN requests to a target, in order to consume resources and cause a DoS. Figure 2.4 depicts an example of a synflood attack carried out in a real computer network. In an interval of ten minutes, more than 210,000 packets are sent as a synflood attack. This network traffic behavior can be considered an abnormal behavior of network traffic, especially since it is concentrated in a short period of time and presents similar outstanding traffic during the time under attack.

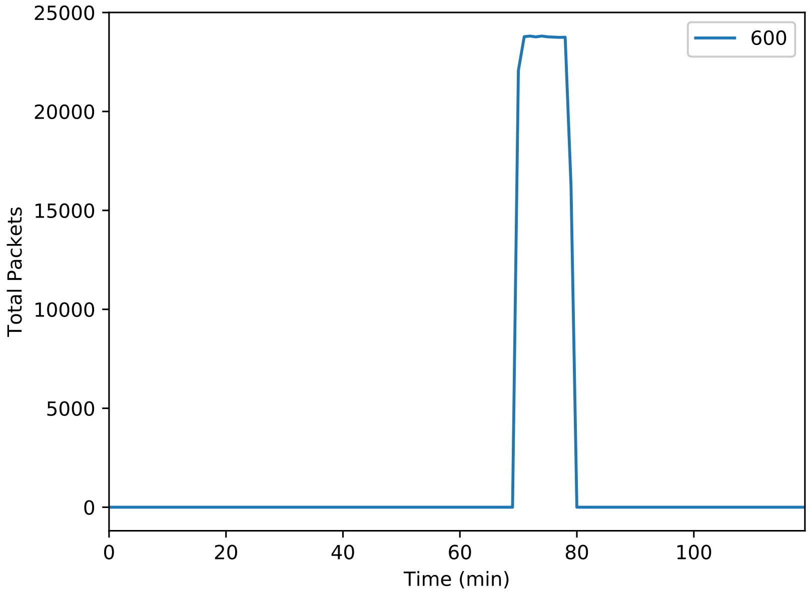


Figure 2.4 A large quantity of SYN requests to a target, in order to cause a DoS.

Regarding the fraggle attack, large packets with UDP echo segments are sent to the broadcast address of a network. Every packet is modified to have the source address of the victim, in order to implement the source address spoofing technique. Therefore, each host receives a huge amount of requests UDP echo and all of them replies to the IP address of the victim, causing a packet flooding aiming a DoS.

The Figure 2.5 depicts an example of the fraggle attack in a real computer network, and shows that more than 6,000,000 malicious packets can be counted in an interval of ten minutes, which can be considered an abnormal network traffic, due to the concentrated traffic in a short period of time and due to the similarity of the outstanding traffic.

The fraggle attack can affect the entire network, since all hosts receive several requests

UDP echo and respond with the Internet Control Message Protocol (ICMP), therefore each host acts as an amplifier of the attack. This part of the fraggle attack is not taken into account in this work, because the victim receives ICMP packets originated from the hosts that are attacked with flooding packet UDP echo.

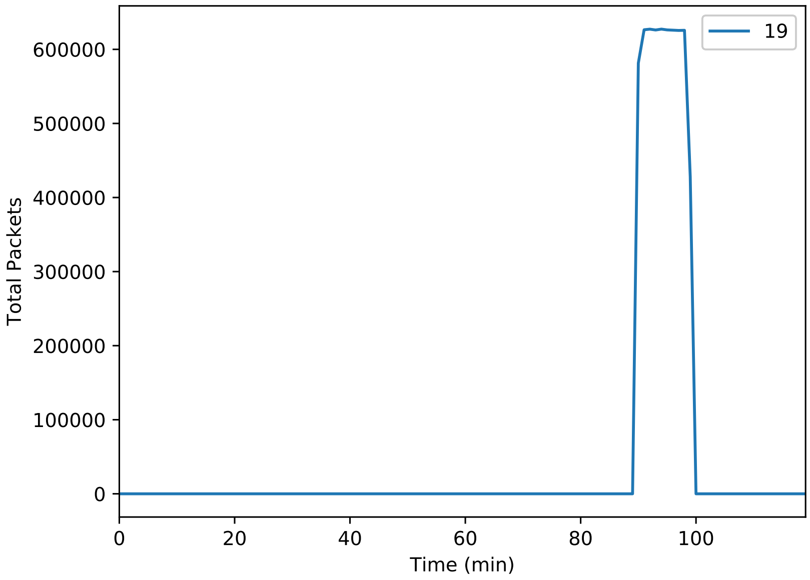


Figure 2.5 Large amount of “UDP echo” requests and replies, causing packet flooding.

Port scan is the attempt to establish a connection to TCP and UDP ports to identify what services are running or are in the listening state. There are several available port scanning techniques, including: TCP SYN scan, TCP ACK scan and UDP scan. This work evaluates the use of TCP SYN scan and UDP scan.

In TCP SYN scan, a SYN packet is sent to the destination and two types of responses may occur: SYN/ACK or RST/ACK. In the first case, the destination port is in the listening state, in the second case, the destination port is not listening. At the end of each port scanning, a RST/ACK packet is sent by the system that is performing the port scan. Therefore, a full connection or a complete three-way handshake is never established, which makes the detection of the attack sender more difficult, and requires approaches able to identify probe attacks without connection establishment.

The UDP scan technique sends UDP packets to the destination port, and if it responds with a *ICMP port unreachable* message, then it indicates that the scanned port is closed. On the other hand, if a message is not received, then the port is considered as open.

Figure 2.6 depicts an example of port scan attack in a real computer network. We simulated a traffic with two packets for each TCP port and one UDP packet to each port.

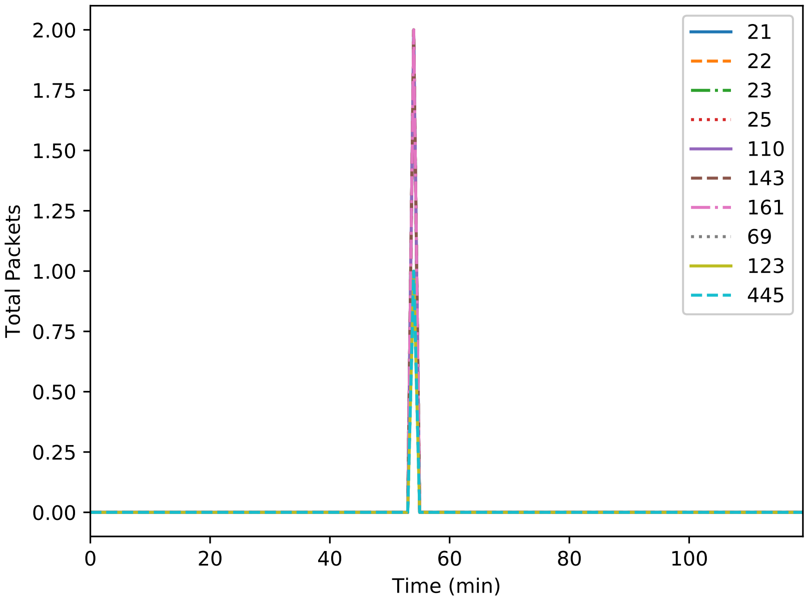


Figure 2.6 Connection attempts in order to identify active ports.

The incoming and outgoing packets analysis, for each port, shows the high correlation and similarity of TCP and UDP traffic during the simulated port scan attack.

2.2.4 The DARPA Data set

The DARPA 1998 data set[4] includes 7 weeks of sniffed traffic saved into raw TCPDUMP packet data, from inside and outside origins, with labeled attacks. The attacks in this data set can be grouped into: denial-of-service (DoS); remote to local (R2L), which is characterized by unauthorized access from a remote machine; user to root (U2R), which is characterized by unauthorized access to local super-user privileges; and probe attack. Since the proposed approach focus on flood and probe attack, the analysis concentrates on the attacks of the DARPA 1998 data set that present behaviors similar to flood or probe attack.

The most cases of DoS of DARPA 98 focus on exploit system vulnerabilities instead of on flooding attack. One example is the occurrence of a neptune attack which sends 20 SYN packets, what is a behavior that differs of the expected flooding attack behavior. Therefore, there were selected the cases that simulates several network traffic or numerous connection requests, also known as flooding attack [20, 36], and the cases that scan ports sending just a few packets. From the simulated probe attacks, we select the cases that

[4] https://www.ll.mit.edu/ideval/data/

rely on TCP or UDP connections.

The data modeling follows the method described by the Subsection 2.2.2, with time frames of 20 minutes, packet aggregation counting by minute and considering the traffic to the following ports: 20, 21, 22, 23, 25, 79, 80, 88, 107, 109, 110, 113, 115, 143, 161, 389, 443.

2.3 Proposed Framework for Detection and Identification of Network Attacks

According to the overview depicted in Figure 2.7, we present in this section the proposed framework for detection and identification of network attacks.

In Subsection 2.3.1 we present the steps for extraction of the largest eigenvalue for each q-th time frame. Next, in Subsection 2.3.2 are presented the mathematical concepts and examples of state-of-the-art MOS schemes, and how to apply the eigenvalues on the MOS scheme in order to detect the attack. In Subsection 2.3.3, we present the eigenvalue analysis to identify the time frames detected as under attack, and the Subsection 2.3.4 describes the similarity analysis evaluated for detailed attack identification.

2.3.1 Largest Eigenvalue by Time Frames

The proposed attack detection algorithm starts by the data preprocessing of a network traffic log containing IP, ports and timestamp of senders and receivers. During this step, the desired information is extracted in order to count packets according to the destination ports by time. Subsequently, this information is grouped by minutes and by time frames.

With the data grouped into Q time frames, the framework is initially applied to each matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$, with $q = 1, \ldots, Q$. According to Subsection 2.2.2, q represents the q-th time frame of aggregated network traffic. The matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$ consists of M network ports and N time bins. We adopt time bins with 1 minute of aggregated traffic and time frames with 20 observations. We assume that the count of ports is defined according to $M < N$, and adopt 17 network ports for our evaluation. Hence, we have $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17 \times 20}$ and each element $x_{m,n}^{(q)}$ stands for the number of packets at n-th minute for the port m, during the q-th time frame.

The time frame size is an important concern to define the sampling size for estimating the sample covariance matrix and the eigenvalue decomposition, considering that if the sample size N is small and the number of variables M is large, the empirical estimators of

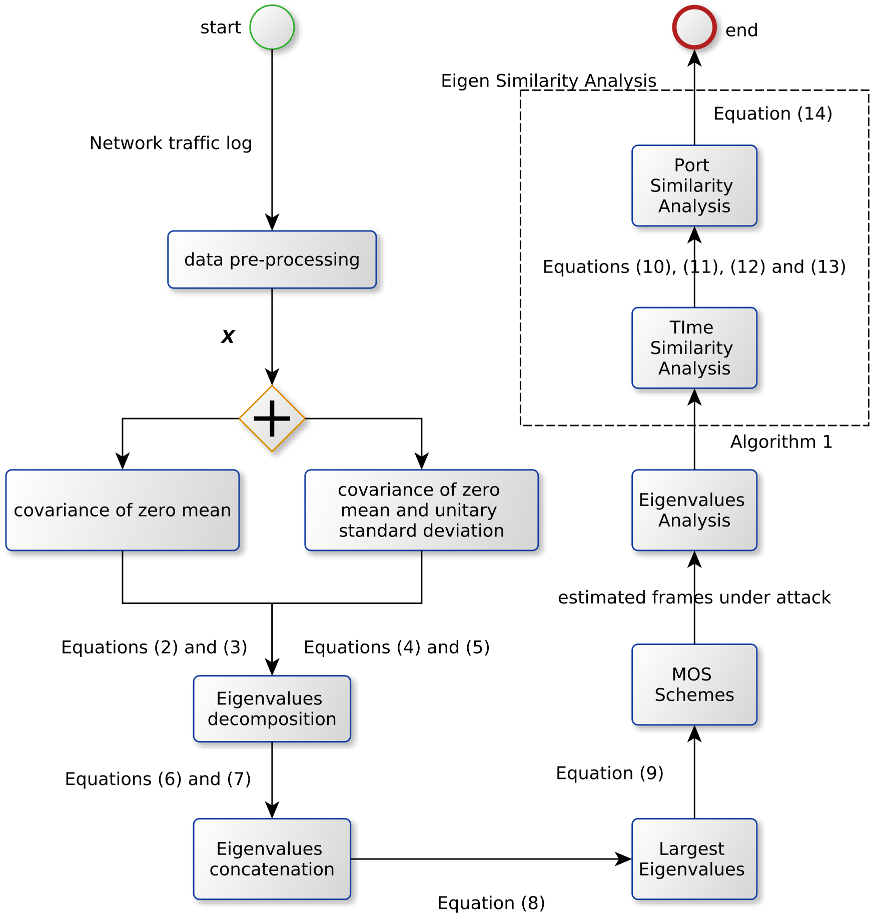


Figure 2.7 Overview of the framework for detection and identification of network attacks.

covariance and correlation can be unstable and the empirical estimate of the covariance matrix becomes singular, i.e. it cannot be inverted to compute the precision matrix.

However, high-dimension, low-sample-size (HDLSS) data are emerging in many areas, such as genetic, imaging, text classification, finance and face recognition. Thus, many methods of shrinkage and regularization have been proposed to improve the stability for estimation of the covariance matrix [47] and eigenvalues [48].

According to flood and port scan attacks' behavior, flood attacks and port scan attacks can be characterized as covariance aware attack [49] and correlation aware attack [11], respectively. These characteristics are substantiated by the results obtained through the analysis based on sample covariance of zero mean variables and on covariance of zero mean and unitary standard deviation variables, described in Section 2.4.

The results in Section 2.4 show that the main components of flood attacks are dominated by the variables with more variance and that the traffic associated with port scan attack does not generate many logs, however, it presents high covariance of zero mean and unitary standard deviation variables.

Therefore, to detect flood attacks, it is necessary to calculate the sample covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ of the zero mean samples given by

$$\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}. \tag{2.2}$$

The set of obtained vectors $\boldsymbol{y}_m^{(q)}$ composes the zero mean matrix $\boldsymbol{Y}^{(q)}$, then the sample covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ can be calculated as follows

$$\hat{\boldsymbol{R}}_{yy}^{(q)} = \frac{1}{N}\boldsymbol{Y}^{(q)}\boldsymbol{Y}^{(q)^{\mathrm{T}}}. \tag{2.3}$$

For the detection of the port scan attack, the main components are not dominated by the variables with large variance. Moreover, the port scan traffic presents a highly correlated network traffic between the monitored ports. In order to exploit such structure, we compute the sample covariance $\hat{\boldsymbol{R}}_{zz}^{(q)}$ whose variables have zero mean and unitary standard deviation as follows

$$\boldsymbol{z}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}}. \tag{2.4}$$

The set of vectors $\boldsymbol{z}_m^{(q)}$ composes the matrix $\boldsymbol{Z}^{(q)}$, then the sample covariance matrix $\hat{\boldsymbol{R}}_{zz}^{(q)}$ can be calculated via

$$\hat{\boldsymbol{R}}_{zz}^{(q)} = \frac{1}{N}\boldsymbol{Z}^{(q)}\boldsymbol{Z}^{(q)^{\mathrm{T}}}. \tag{2.5}$$

Once the $\hat{\boldsymbol{R}}_{yy}^{(q)}$ and $\hat{\boldsymbol{R}}_{zz}^{(q)}$ have been obtained for flood and port scan detection, respectively, and since the next steps are the same for both sample covariance matrices, we refer to $\hat{\boldsymbol{R}}_{yy}$ and $\hat{\boldsymbol{R}}_{zz}$ as a matrix $\hat{\boldsymbol{R}}$. Therefore, the following step of the algorithm is the eigenvalue decomposition (EVD), calculated according to (2.6), in order to obtain the vector of eigenvalues $\boldsymbol{e}^{(q)}$ associated with each matrix, according to (2.6).

$$\hat{\boldsymbol{R}}^{(q)} = \boldsymbol{V}^{(q)}\boldsymbol{\Lambda}^{(q)}\boldsymbol{V}^{(q)^{\mathrm{T}}}, \tag{2.6}$$

$$\boldsymbol{e}^{(q)} = \mathrm{diag}(\boldsymbol{\Lambda}^{(\mathrm{q})}), \tag{2.7}$$

where the operator $\mathrm{diag}(\cdot)$ extracts the main diagonal of a matrix.

The eigenvalues should be sorted in descending order, i.e., $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > ... >$

$\lambda_m^{(q)}$. Therefore, the largest eigenvalue of the q-th time frame evaluated for the attack detect is given by $\lambda_1^{(q)}$.

The concatenation of the eigenvalues vector $\boldsymbol{e}^{(q)}$ for $q = 1, \ldots, Q$ is represented by

$$\boldsymbol{E} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}. \tag{2.8}$$

Note that since $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, then the first line of the matrix $\boldsymbol{E}$ contains the largest eigenvalues of each q-th time frame, which is the Greatest Eigenvalue Time Vector (GETV) [35], denoted as

$$\boldsymbol{e}_{\max} = [\lambda_1^{(1)}, \lambda_1^{(2)} \ldots \lambda_1^{(Q)}] \tag{2.9}$$

2.3.2 Model Order Selection (MOS)

The model order selection is a key point in many digital signal processing applications, including radar, sonar, communications, channel modeling, medical imaging, among others [50, 51, 52]. MOS allows analysis of reduced data set, through separating noise components of the main components, for example. Moreover, the model order is crucial for many parameter estimation techniques [53, 54], since the amount of parameters to be estimated depends on the model order.

The model selection procedure chooses the “best” model of a finite set of models, according to some criteria [55]. Therefore, given some data set, it is chosen a model which was evaluated as the best model to describe the specified data set.

The state of the art regarding estimation techniques of model order based on eigenvalues includes: Akaike’s Information Theoretic Criterion - AIC [56, 57]; Minimum Description Length - MDL [58, 57]; Efficient Detection Criterion - EDC [59]; Stein’s Unbiased Risk Estimator - SURE [60]; RADOI [61] and Exponential Fitting Test - EFT [62, 63, 33].

In AIC, MDL and EDC techniques, the information criterion is a function of the geometric mean $g(k)$ and the arithmetic mean $a(k)$ relating to smaller k eigenvalues, where k is a candidate value for the model order d [54].

Basically, the difference between the AIC, MDL and EDC schemes is the penalty function $p(k,N,\alpha)$, so these techniques can be written in general as [54]:

$$\hat{d} = \arg\min_{k} J(k), \tag{2.10}$$

where

$$J(k) = -N(\alpha - k)\log\left(g(k)/a(k)\right) + p(k,N,\alpha), \tag{2.11}$$

where $\hat{d}$ is an estimate d of the model order, N is the number of samples, $\alpha = M$ and means the number of variables of the problem, and $0 \leqslant k \leqslant min[M,N]$.

Penalty functions for AIC, MDL and EDC are given by the Table 2.1.

Table 2.1 Penalty functions for the schemes AIC, MDL and EDC

Scheme	**Penalty function** $p(k,N,\alpha)$
AIC	$k(2\alpha - k)$
MDL	$0.5k(2\alpha - k)\log(N)$
EDC	$0.5k(2\alpha - k)\sqrt{N\,\mathrm{ln(lnN)}}$

The Exponential Fitting Test (EFT) can effectively be used in cases where the number of samples N is small. This technique is based on observations of data contaminated only with white noise, where the profile of eigenvalues can be approximated by an exponential decaying [62].

Given λ_i be the i-th eigenvalue, the exponential model can be expressed by:

$$E\{\lambda_i\} = E\{\lambda_1\} \cdot q(\alpha,\beta)^{i-1}, \tag{2.12}$$

where $E\{\cdot\}$ is the expectation operator, and it is considered that the eigenvalues are ordered in the that λ_1 represents the largest eigenvalue. The term $q(\alpha,\beta)$ is defined as:

$$q(\alpha,\beta) = \exp\left\{-\sqrt{\frac{30}{\alpha^2+2} - \sqrt{\frac{900}{(\alpha^2+2)^2} - \frac{720\alpha}{\beta(\alpha^4+\alpha^2-2)}}}\right\}, \tag{2.13}$$

where $0 < q(\alpha,\beta) < 1$. According to [63], if $M \leq N$, then $\beta = N$.

Traditionally, the MOS schemes are applied for the eigenvalues, that are the vector $\boldsymbol{e}^{(q)}$ when considering the application of MOS schemes for Eigensimilarity. However, the goal of our proposal is to detect the variations of the eigenvalues for different values of q.

Instead of using a certain q, the proposed approach applies MOS schemes for a vector of the largest eigenvalues of each q-th time frame, in order to identify variations and estimate the model order $\hat{d}$, which is the estimated number of time frames under attack. Therefore, $\boldsymbol{e}_{\max}$ is sorted in descending order, producing $\sim \boldsymbol{e}_{\max}$, that is used as input parameter for MOS schemes, according to $\hat{d} = \mathrm{MOS}(\sim \boldsymbol{e}_{\max})$. Note that some MOS schemes may also require the number of minutes that compose a time frame, as $\hat{d} = \mathrm{MOS}(\boldsymbol{e}_{\max}, Q)$.

In our previous work [35], the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE schemes are evaluated for synflood and port scan attack detection, showing that EDC and EFT are effective for detecting this kind of attacks. The present work extends that evaluation to also analyze the effectiveness of the listed MOS schemes for fraggle attack detection, as shown in Section 2.4.

2.3.3 Eigenvalue Analysis

After applying the MOS schemes to the vector $\sim \boldsymbol{e}_{\max}$, we obtain the rank estimate of the $\#\boldsymbol{X}$. For instance, in the case of fraggle, synflood and ports can, if $\hat{d} = 1$, then $\#\boldsymbol{X} = 1$ indicate a estimate of rank 1 for $\boldsymbol{X}$, which means that during the during the Q time frames one time frame q is under attack. However, if $\hat{d} = 0$, then $\#\boldsymbol{X} = 0$, and this means that none of the Q time frames is under attack. Note that $\hat{d}$ can be greater than 1, indicating the presence of more than one attacked time frame.

In Subsection 2.3.2, we obtained only if $\hat{d} = 1$ or $\hat{d} = 0$, estimating the number of time frames under attack. However, if $\hat{d} > 1$, the MOS schemes does not identify the q-th attacked time frames. The identification of the q-th time frame under attack can be carried out through an eigenvalues analysis.

The largest eigenvalue analysis for estimating the q-th time frames that are under attack can be expressed according to Algorithm 1, where $\hat{\boldsymbol{q}}_{\max} \in \mathbb{R}^{\hat{\mathrm{d}}}$ denotes a vector of the q-th time frames under attack, which is the q-th indexes corresponding to the $\hat{d}$ largest eigenvalues of $\boldsymbol{e}_{\max}$. Algorithm 1 initially identifies the largest value of $\boldsymbol{e}_{\max}$, according to Line 3 of Algorithm 1, and its correspondent index, according to Line 7. Subsequently, the largest value is removed of $\boldsymbol{e}_{\max}$, according to Line 11 of Algorithm 1, and a new iteration is performed until $\boldsymbol{e}_{\max} = []$.

After the estimation of the $\hat{\boldsymbol{q}}_{\max}$ time frames under attack, it is necessary to obtain more details of the detected Attacks, such as the n-th minutes when the attacks happened and the m-th network ports that were attacked. To deal with this problem, the adoption of

Algorithm 1: Detection of Time Frames Under Attack

```
Result: $\hat{\boldsymbol{q}}_{\max}$
Given $f = 1$;
while $f < \hat{d}$ do
    $q_{\text{value}} = \text{argmax}_{\lambda}\ \boldsymbol{e}_{\max}$;
    $i = 1$;
    while $i < Q$ do
        if $\boldsymbol{e}^{(i)}_{\max} == q_{\text{value}}$ then
            $\hat{\boldsymbol{q}}^{(f)}_{\max} = i$;
        end
        $i = i + 1$;
    end
    $\boldsymbol{e}_{\max} \rightarrow \hat{\boldsymbol{q}}^{(f)}_{\max}$;
    $f = f + 1$;
end
```

a similarity analysis between legitimate traffic and the traffic of time frames estimated as under attack is evaluated, analyzing the effectiveness of cosine similarity to highlight abnormalities inserted by network traffic attacks.

2.3.4 Principal Eigenvector Similarity Analysis

The eigenvector corresponding to the eigenvalue of largest magnitude is called the principal eigenvector or dominant eigenvector. Cosine similarity calculates the cosine of the angle between two vectors, which represents the similarity of values between the selected vectors. Therefore, cosine similarity can be used to evaluate the difference between the principal eigenvector $\boldsymbol{V}^{(q)}$ against the principal eigenvector of one time frame detected as under attack, in order to analyze similarity changes between the principal eigenvectors with the largest variance caused by the insertion of anomalous traffic [42] in a time frame.

The cosine similarity is a measure between two vectors and does not consider the distance between points or features of a vector. Therefore, the computed angle between eigenvectors does not reveal element-wise distance neither the contribution of each component for the measured distance, but can be used to identify the principal eigenvector of observations that deviates from a reference principal eigenvector and can identify network attacks. After the identification of the time under attack by means of the cosine similarity between principal eigenvectors, the cosine similarity can also be applied for identification of components that insert the deviation between the principal eigenvectors, according explained in subsection 2.3.4.2.

This subsection describes the proposed principal eigenvector similarity analysis for detailed attack identification, in complement to the attack estimation carried out through MOS schemes and eigenvalue analysis. In Subsection 2.3.4.1 we present the principal eigenvector similarity analysis for identification of time under attack. Next, in Subsection 2.3.4.2, we show how to apply the principal eigenvector similarity analysis in order to identify network ports under attack.

2.3.4.1 Time Similarity Analysis

For principal eigenvector similarity analysis, we evaluate the cosine similarity in order to identify lacks of similarity between legitimate and malicious traffic, as follows:

$$s_n = \frac{|\boldsymbol{v}^{(q)} \cdot \boldsymbol{v}_{(n)}^{(q)}|}{||\boldsymbol{v}^{(q)}|| \, ||\boldsymbol{v}_{(n)}^{(q)}||}, \tag{2.14}$$

where s_n denotes the absolute similarity degree between the n-th minute and the reference principal eigenvector, given by an inner product that measures the cosine of the angle between two vectors, where the l_2 norm is denoted by $||\cdot||$ and $|\cdot|$ denotes the absolute value.

In (2.14), the $\boldsymbol{v}^{(q)}$ denotes the principal eigenvector of a selected set of minutes without network attack, according the attack detection described by Algorithm 1, and $\boldsymbol{v}_{(n)}^{(q)}$ is the principal eigenvector obtained after append each target n-th minute of traffic that needs the identification of flood and port scan attacks.

The principal eigenvector $\boldsymbol{v}^{(q)}$ of a time frame q without attack can be computed from (2.6) and selected according to the eigenvector corresponding to the largest eigenvalue $\lambda_1^{(q)}$, which is the principal component of the selected time frame q. The same calculation shall be performed in order to obtain the target principal eigenvector $\boldsymbol{v}_{(n)}^{(q)}$, calculated from the time frame without attack appended by selected minutes of a time frame estimated as under attack.

Therefore, the principal eigenvector $\boldsymbol{v}^{(q)}$ is calculated from the traffic without attack, in a time frame q composed of Q minutes of legitimate network traffic, estimated as normal by Algorithm 1. For the detailed attack identification, each $\boldsymbol{x}_{(n)}^{(\hat{q})}$ vector of each n-th minutes of the estimated $\hat{\boldsymbol{q}}_{\max}$ time frames shall be individually appended into $\boldsymbol{X}^{(q)}$, as represented by

$$\boldsymbol{X}_n = \{\boldsymbol{X}^{(q)} | \boldsymbol{x}^{(\hat{q})}_{(n)}\}. \tag{2.15}$$

Subsequently, the resultant $\boldsymbol{X}_{(n)}$ is used to obtain $\boldsymbol{v}^{(q)}_{(n)}$, through (2.6), for calculating the similarity degree s_n, ranging from 0 to 1, for each n-th minute. The s_n denotes the absolute similarity degree of the n-th minute in comparison to a well-known traffic without attack, detected through MOS schemes and eigenvalue analysis.

We propose three approaches for principal eigenvector similarity analysis, which are the incremental, the individual and the incremental individualized.

The incremental approach for principal eigenvector similarity analysis is based on the incremental appending of network traffic into $\boldsymbol{X}^{(q)}$, where the first evaluation is based on (2.15) and the subsequent evaluations are based on (2.16), that denotes the appending of the n-th minute $\boldsymbol{x}^{(\hat{q})}_{(n)}$ into $\boldsymbol{X_n}$. The incremental approach repeats the (2.16) while $n \leq N$ and compute $\boldsymbol{v}^{(q)}$, $\boldsymbol{v}^{(q)}_{(n)}$ and the (2.14) for each increment.

$$\boldsymbol{X}_n = \{\boldsymbol{X}_n | \boldsymbol{x}^{(\hat{q})}_{(n)}\}, \tag{2.16}$$

Figure 2.8 illustrates the network traffic selection for the incremental approach of principal eigenvector similarity analysis, where the $\boldsymbol{X}^{(1)}$ is chosen as reference for similarity analysis of the m-th minutes of the time frame $q = 3$, where one network attack was previously detected by Algorithm 1.

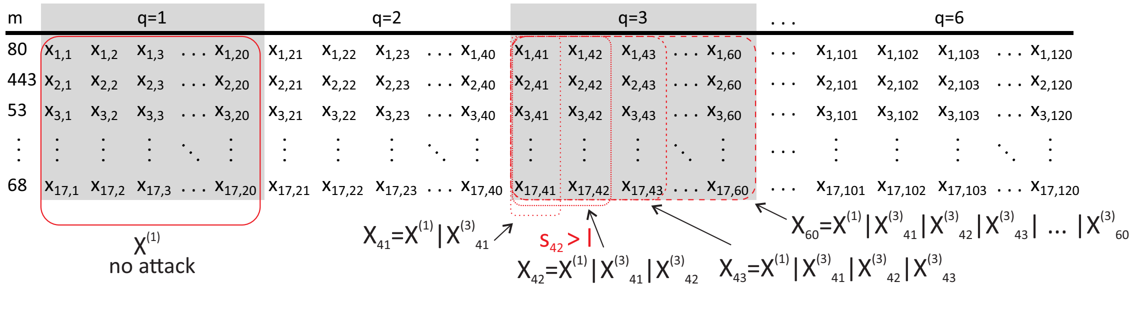

m
q=1
q=2
q=3
...
q=6
80
443
53
68
x1,1 x1,2 x1,3 ... x1,20
x2,1 x2,2 x2,3 ... x2,20
x3,1 x3,2 x3,3 ... x3,20
x17,1 x17,2 x17,3 ... x17,20
x1,21 x1,22 x1,23 ... x1,40
x2,21 x2,22 x2,23 ... x2,40
x3,21 x3,22 x3,23 ... x3,40
x17,21 x17,22 x17,23 ... x17,40
x1,41 x1,42 x1,43 ... x1,60
x2,41 x2,42 x2,43 ... x2,60
x3,41 x3,42 x3,43 ... x3,60
x17,41 x17,42 x17,43 ... x17,60
x1,101 x1,102 x1,103 ... x1,120
x2,101 x2,102 x2,103 ... x2,120
x3,101 x3,102 x3,103 ... x3,120
x17,101 x17,102 x17,103 ... x17,120
X(1)
no attack
X41=X(1)|X(3)41
s42 > l
X42=X(1)|X(3)41|X(3)42
X43=X(1)|X(3)41|X(3)42|X(3)43
X60=X(1)|X(3)41|X(3)42|X(3)43| ... |X(3)60

Figure 2.8 Traffic selection for incremental approach.

For the scenario depicted by Figure 2.7, the principal eigenvector similarity analysis starts at $\boldsymbol{x}^{(3)}_{(41)}$ and is incrementally performed until $\boldsymbol{x}^{(3)}_{(60)}$, in order to calculate the s_n. We assume that $s_n < l$ means an attack identification, according the anomaly on similarity of s_n compared to a defined threshold l.

Therefore, after obtaining the principal eigenvector $\boldsymbol{v}^{(q)}$ and the target principal eigenvector $\boldsymbol{v}_{(n)}^{(q)}$ for principal eigenvector similarity analysis, the s_n is calculated according to (2.14). If $s_n = 1$, then the two principal eigenvectors are completely similar and no anomaly is detected. Smaller values of s_n mean less similarity and can indicate an anomaly if $s_n < l$, what denotes that a network attack is identified during the n-th minute.

The (2.17) shows how the s_n of each n-th minute shall be compared with the threshold l to evaluate if an attack is identified, where

$$\hat{\boldsymbol{n}}_{(n)} = \begin{cases} 1, & \text{if } s_n < l \\ 0, & \text{otherwise} \end{cases}, \tag{2.17}$$

and $\hat{\boldsymbol{n}}_{(n)}$ denotes a vector of n-th minutes detected as under attack.

The principal eigenvector similarity analysis can also be applied by means of the individual approach, where each n-th minute must be individually appended into $\boldsymbol{X}^{(q)}$, as shown by Figure 2.9. In the individual approach there is no incremental appending, therefore only individual n-th minute are appended to $\boldsymbol{X_n}$, according to (2.15), in order to compute $\boldsymbol{v}^{(q)}$, $\boldsymbol{v}_{(n)}^{(q)}$ and the (2.14) for individual n-th minute.

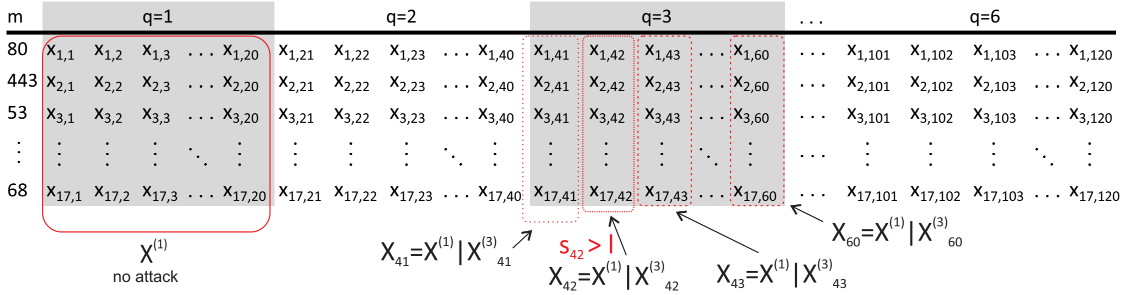

m
q=1
q=2
q=3
...
q=6
80
443
53
⋮
68
x1,1 x1,2 x1,3 ... x1,20 x1,21 x1,22 x1,23 ... x1,40 x1,41 x1,42 x1,43 ... x1,60 ... x1,101 x1,102 x1,103 ... x1,120
x2,1 x2,2 x2,3 ... x2,20 x2,21 x2,22 x2,23 ... x2,40 x2,41 x2,42 x2,43 ... x2,60 ... x2,101 x2,102 x2,103 ... x2,120
x3,1 x3,2 x3,3 ... x3,20 x3,21 x3,22 x3,23 ... x3,40 x3,41 x3,42 x3,43 ... x3,60 ... x3,101 x3,102 x3,103 ... x3,120
x17,1 x17,2 x17,3 ... x17,20 x17,21 x17,22 x17,23 ... x17,40 x17,41 x17,42 x17,43 ... x17,60 ... x17,101 x17,102 x17,103 ... x17,120
X(1)
no attack
X41=X(1)|X(3)41
S42 > l
X42=X(1)|X(3)42
X43=X(1)|X(3)43
X60=X(1)|X(3)60

Figure 2.9 Traffic selection for individual approach.

The incremental and the individual approaches can be combined to obtain the incremental individualized approach, where each minute is incrementally appended into the selected $\boldsymbol{X}^{(q)}$ for obtaining $\boldsymbol{v}_{(n)}^{(q)}$ to compute similarity analysis of the n-th minute, until detect the first n-th minute under attack, i.e. $s_n < l$. Subsequently, $\boldsymbol{X}_{n-1}$ becomes the new reference of traffic without network attack and each subsequent minute must have its similarity individually evaluated, as shown in Figure 2.10.

The incremental similarity analysis followed by individual analysis after an attack detection allows to identify the attack period, highlighting the first and last time under attack. This identification is possible due to the similarity variation between the principal eigenvectors, which highlight lacks of similarity when compared the principal eigenvector of a traffic under attack against the principal eigenvector of a traffic with no attack,

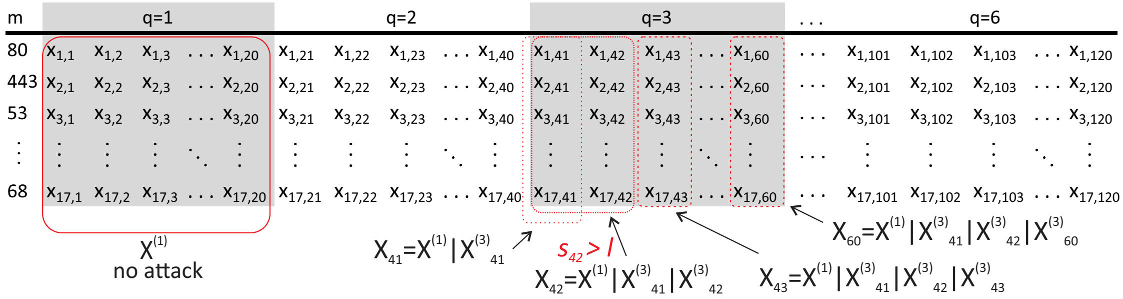


Figure 2.10 Traffic selection for incremental individualized approach.

according to results which are discussed in Section 2.4.

2.3.4.2 Port Similarity Analysis

Given $\hat{\boldsymbol{n}}$, which is the set of n-th minutes under attack, it is still necessary to obtain more details about the identified network attack, such as the network ports that are attacked during each n-th minute identified as under attack. Hence, it is also applied the cosine similarity analysis to identify variation of the principal eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port during a n-th minute.

For detection of ports under attack, the last principal eigenvector without attack $\boldsymbol{v}^{(q)}$ shall be used as reference for similarity analysis against the $\boldsymbol{v}^{(q)}_{(n)}$ identified as under attack, and evaluate individually the cosine similarity of each m-th port of all $\hat{\boldsymbol{n}}$ minutes. Therefore, $\boldsymbol{v}^{(q)}$ should be calculated from the last $\boldsymbol{X}^{(q)}$ time frame without attack, and $\boldsymbol{v}_{(m,\hat{n})}$ should be calculated from the same traffic appended of all n-th minutes until the identified minute under attack, denoted as $\boldsymbol{X}_n$.

For similarity analysis, each m-th port of the last n-th minute of $\boldsymbol{X}_n$, denoted as $x_{(m,n)}$, shall be individually replaced by the traffic of the evaluated m-th port of the $\hat{n}$-th minute under attack, denoted as $x^{(\hat{q})}_{(m,\hat{n})}$, in order to identify significant variation on similarity caused by the traffic of the m-th port.

This approach for detection of ports under attack via similarity analysis is given by

$$\begin{cases} x_{(m,n)} = x^{(\hat{q})}_{(m,\hat{n})} \\ \\ s_{m,\hat{n}} = \frac{|\boldsymbol{v}^{(q)} \cdot \boldsymbol{v}_{(m,\hat{n})}|}{\|\boldsymbol{v}^{(q)}\| \|\boldsymbol{v}_{(m,\hat{n})}\|}, \end{cases} \tag{2.18}$$

where $x^{(\hat{q})}_{(m,\hat{n})}$ denotes the m-th port of the selected n-th minute and q-th time frame identified as under attack and $x_{(m,n)}$ denotes the m-th port of the last n-th minute of $\boldsymbol{X}_n$, which is used to calculate the $\boldsymbol{v}_{(m,\hat{n})}$ principal eigenvector that contains the traffic of the m-th port of the $\hat{n}$-th minute identified as under attack.

Once $\boldsymbol{v}^{(q)}$ and $\boldsymbol{v}_{(m,\hat{n})}$ are obtained, then the $s_{m,\hat{n}}$ similarity degree can be calculated in order to identify if the traffic replacement highlights the addition of anomalous traffic by the evaluated m-th port during the $\hat{n}$-th minute previously identified as under attack.

This procedure should be repeated for each m-th target port of $\hat{\boldsymbol{n}}$, in order to individually identify the network ports under attack during each $\hat{q}$-th time frame.

2.4 Experiments and Results

This section presents the performed experiments and the acquired results for the Eigensimilarity. First, in Section 2.4.1, the experimental scenario adopted in the evaluation is summarized. Then, in Section 2.4.2 the results of the largest eigenvalue analysis by time frames for the experimental scenario are shown. In Section 2.4.3 we describe the results of the evaluated MOS schemes for attack detection in the simulated data set. In Section 2.4.4 we present the results of the eigenvalue analysis for identification of time frames under attack. In Section 2.4.5 we show the results of similarity analysis for detailed flood and port scan identification for the experimental scenario. In Section 2.4.6 we present the results of the proposed framework for flood and probe attack detection in the DARPA 1998 data set.

2.4.1 Experimental Scenario

This experiment considers a simulated scenario of a real network monitored during 120 minutes, that are separated into six time frames of twenty minutes. Therefore, as the time of each sampling period is one minute, then $N = 20$. For each time frame q, a traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17\times 20}$ was obtained, as well as a covariance $\hat{\boldsymbol{R}}_{yy}^{(q)} \in \mathbb{R}^{17\times 17}$ (calculated via (2.3)) and a sample covariance matrix $\hat{\boldsymbol{R}}_{zz}^{(q)} \in \mathbb{R}^{17x17}$, assuming that $q = 1, 2, 3, 4, 5$ and 6.

The simulation started at 21:00h, the first time frame was from 21:00h until 21:20h ($q = 1$), the second was from 21:20h until 21:40h ($q = 2$), the third was from 21:40h to 22:00h ($q = 3$), the fourth was from 22:00h until 22:20h ($q = 4$), the fifth was from 22:20h until 22:40h ($q = 5$), and finally, the sixth was from 22:40h until 23.00h ($q = 6$). During the simulation, the victim made legitimate access, and the attacker performed the following attacks: at 21:54h ($q = 3$) was performed a port scan, at the interval ranging from 22:10h to 22:20h ($q = 4$) a synflood attack was simulated, and at the interval from 22:30h to 22:40h ($q = 5$) a fraggle attack was performed.

2.4.2 Largest Eigenvalues Analysis

For the evaluation of MOS Schemes accuracy for flood and port scan detection, the framework defines that it is necessary to obtain the largest eigenvalue of each time frame, through eigen decomposition from a covariance of zero mean variables or covariance matrix of zero mean and unitary standard deviation variables, calculated from the evaluated traffic, as described in Section 2.3.

Through eigenvalue analysis of traffic with flood or port scan attacks, it is possible to visualize a significant difference between the largest eigenvalues and the remain eigenvalues, which can indicate a relationship between an outlier and time frames under attack.

Figure 2.11 depicts the eigenvalues calculated from sample covariance matrix of the network traffic used to evaluate the synflood attack identification. In Figure 2.11, the largest eigenvalue related to the simulated synflood attack ($q = 4$) stands out significantly from the other eigenvalues.

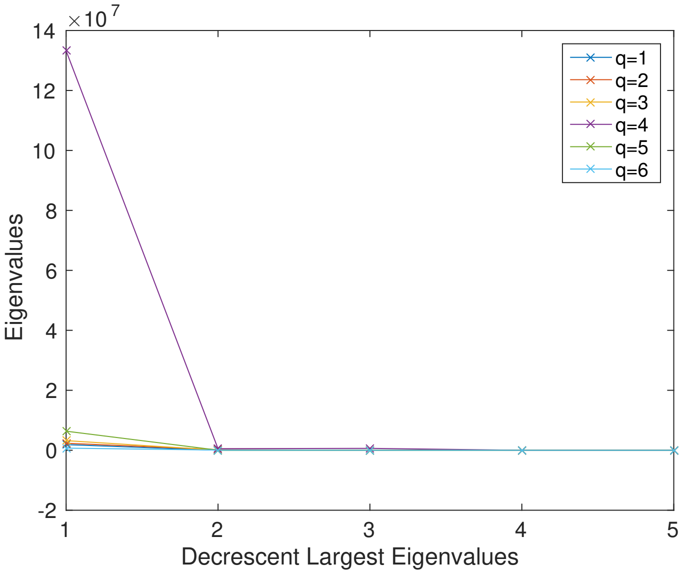


Figure 2.11 Eigenvalues of the sample covariance matrix (synflood).

Figure 2.12 illustrates the eigenvalues calculated from sample covariance matrix of the matrix used for fraggle attack detection. In Figure 2.11, the largest eigenvalue related to the simulated synflood attack ($q = 5$) stands out significantly from the other eigenvalues, in accordance with the result shown in Figure 2.11 for the synflood attack analysis.

Figure 2.13 depicts the eigenvalues calculated from covariance matrix of zero mean

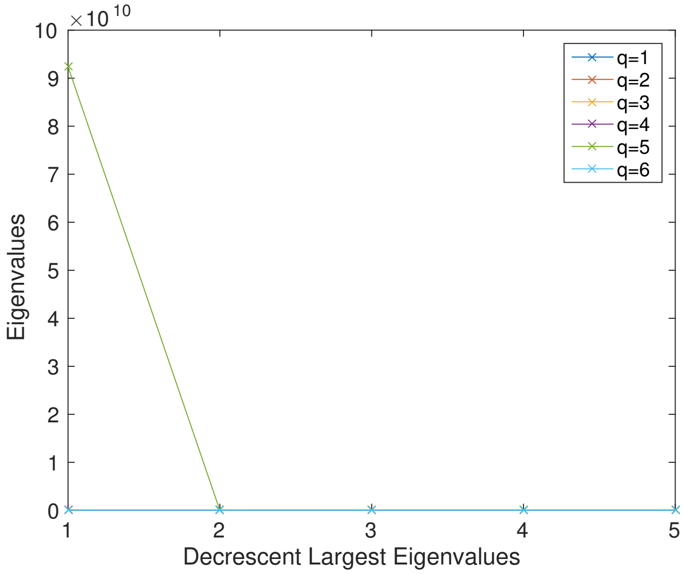


Figure 2.12 Eigenvalues of the sample covariance matrix (fraggle).

and unitary standard deviation variables, of the network traffic matrix evaluated for port scan detection. As analyzed for the synflood and fraggle attacks, note that the largest eigenvalue, related to this attack ($q = 3$), stands out significantly from the others eigenvalues.

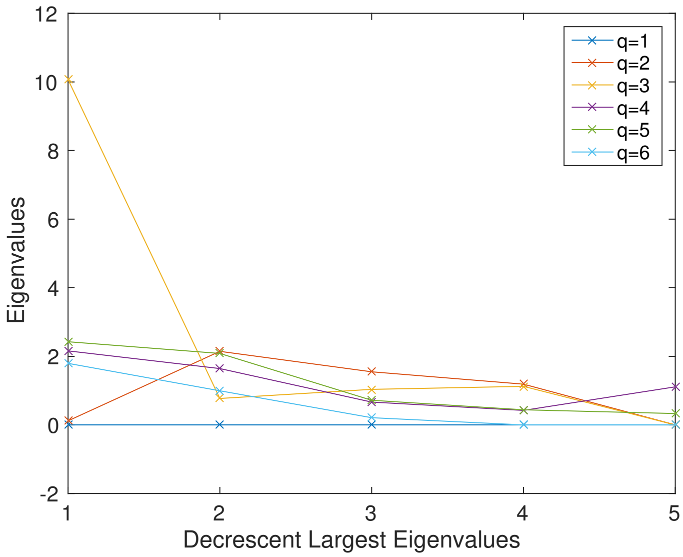


Figure 2.13 Eigenvalues of the covariance matrix of zero mean and unitary standard deviation (port scan).

Table 2.2 presents the values of the largest eigenvalues of each time frame q-th for port scan, synflood and fraggle detection.

Table 2.2 Largest Eigenvalue related to attacks detection

Time Frame q	Vectors GETV			
	Detection of ***synflood/fraggle***	**Detection of** ***synflood***	**Detection of** ***fraggle***	**Detection of** ***port scan***
1	1887545	1887545	1887545	2,0734
2	2341327	2341327	2341327	2,1451
3	3213867	3213867	3213867	10,0718
4	133238294	133238294	731229	2,1620
5	92384021611	6367983	92384021611	2,4253
6	708335	708335	708335	1,7948

In Table 2.2, note the significant variation of the eigenvalues associated with attacks, in comparison to the others. At $q = 4$, where the synflood attack occurred, the maximum eigenvalue obtained is approximately 21 times larger than the second one. At $q = 5$, where the fraggle attack occurred, the maximum eigenvalue obtained is about 29,000 times larger than the second one. At $q = 3$, where the port scan attack occurred, the maximum eigenvalue obtained is approximately 4 times larger than the second one. In the last case, for port scan attack detection, although the largest eigenvalue presented no too large variance to the second one, if compared to synflood or fraggle attacks, it clearly deviates from the remaining largest eigenvalues.

These results highlight that all q-th time frames where a network attack was simulated, present high significant variance between the largest eigenvalue and the remaining eigenvalues, obtained from sample covariance matrix, for flood detection, or from covariance matrix of zero mean and unitary standard deviation variables, for port scan detection. Therefore, we propose to apply the vector of the largest eigenvalues to MOS schemes in order to evaluate their accuracy for identification of time frames under attack, motivated by the fact that it is relevant to apply MOS schemes to automate the attack detection process, taking into account the characteristics of the evaluated eigenvalues.

2.4.3 MOS Schemes Evaluation

In [35], we evaluate the accuracy of AIC, MDL, EDC, RADOI, EFT and SURE MOS schemes [54, 35] for synflood and port scan attack detection. In this work we extend that evaluation for fraggle attack detection, applying the same schemes to fraggle attack detection over the traffic presented in Section 2.2, as results shown in Table 2.3.

Note that $\hat{d} = 1$, if there is one attack, while $\hat{d} > 1$ indicates more than one attack. An example of this could be seen for attack detection via EFT for traffic containing synflood and fraggle attacks, showing $\hat{d} = 2$, which indicates the presence of two attacks, as expected by the ground truth values *d* of Table 2.3.

Table 2.3 MOS schemes applied to port scan and flood detection

Type of analysis q	MOS schemes (estimated model order $\hat{d}$)						(d)
	AIC	**MDL**	**EDC**	**RADOI**	**EFT**	**SURE**	
Detection of synflood (presence of attack)	2	1	**1**	5	**1**	4	**1**
Detection of synflood (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of fraggle (presence of attack)	1	1	**1**	5	**1**	4	**1**
Detection of fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**
Detection of port scan (presence of attack)	1	1	**1**	1	**1**	9	**1**
Detection of port scan (absence of attack)	0	0	**0**	1	**0**	1	**0**
Detection of synflood/fraggle (presence of attack)	2	2	**2**	5	**2**	5	**2**
Detection of synflood/fraggle (absence of attack)	1	1	**0**	1	**0**	3	**0**

In Table 2.3, two MOS schemes outperforms from the others, which are EDC and EFT. Efficient Detection Criterion (EDC) and Exponential Fitting Test (EFT) are the most effective schemes, correctly estimating the number of attacks in comparison to the expected values for effective attack detection, as defined by the column of real values in Table 2.3. The AIC and MDL schemes are satisfactory only for port scan detection, however SURE and RADOI schemes did not show effective results for port scan or flood detection.

Although EDC and EFT presented the same accuracy on the evaluation, the EDC scheme requires less processing time than EFT, which is an important criteria to select EDC as the MOS scheme for flood and port scan detection on the remain experiments.

According to Table 2.3, EDC and EFT estimated correctly the number of attacks of a time frame vector, indicating that occurred $\hat{d}$ network attacks, but not providing additional details, what highlights the necessity of complementary approaches in order to estimate the time and ports under attack. Hence, we propose apply eigen analysis to estimate the q-th time frames under attack and principal eigenvector similarity analysis to estimate the minutes and ports under attack.

2.4.4 Eigenvalue Analysis

According to the results presented in Section 2.4.2, the largest eigenvalue stands out significantly from the others eigenvalues of an evaluated q-th time frame. This behavior can also be observed in the largest eigenvalues analysis, according to results presented in Table 2.2, where it is possible to observe that the $\hat{d}$ largest eigen values of the time frames under attacks stand out significantly from the others largest eigenvalues.

Therefore, we conclude that the $\hat{d}$ largest eigenvalues correspond to the respective q-th time frames under attack, which is denoted by $\hat{\boldsymbol{q}}_{\max}$ and can be calculated according to Algorithm 1.

2.4.5 Principal Eigenvector Similarity Analysis

In order to analyze the hypotheses $H_1^{(N)}$ and $H_1^{(A)}$, which evaluate if a subspace learned by eigenvalue decomposition can be used to detect and identify network attacks, we propose to apply principal eigenvector similarity analysis to detect time and ports under attack, from each q-th time frames under attack defined by $\hat{\boldsymbol{q}}_{\max}$. Hence, the proposed framework is applied to the time frames where $q = 3$, $q = 4$ and $q = 5$ to respectively evaluate its effectiveness for port scan, synflood and fraggle attack detection.

2.4.5.1 Time Analysis

Three approaches were evaluated for principal eigenvector similarity analysis: incremental, individual and incremental individualized approaches. For the incremental individualized approach, each minute is incrementally appended into the selected $\boldsymbol{X}^{(q)}$ for obtaining $\boldsymbol{v}_{(n)}^{(q)}$ to similarity analysis of the n-th minute, until detect the first n-th minute under attack. Subsequently, $\boldsymbol{X}_n$ became the new reference of traffic without network attack and each subsequent minute must have its similarity individually evaluated. For the incremental approach, each n-th minute must be incrementally appended into $\boldsymbol{X}^{(q)}$, for obtaining the next principal eigenvector $\boldsymbol{v}_{(n)}^{(q)}$ for individual time similarity analysis. For the individual approach, each n-th minute must be individually appended into $\boldsymbol{X}^{(q)}$, without incremental append, but doing individual appended into $\boldsymbol{X}^{(q)}$ for obtaining the next principal eigenvector $\boldsymbol{v}_{(n)}^{(q)}$ for individual similarity analysis.

Table 2.4 presents the results of the evaluation of three approaches for similarity analysis of principal eigenvectors for port scan detection. Table 2.4 shows the evaluation of the time frame $q = 3$, when the port scan attack was simulated, considering the incremental individualized, incremental and individual approaches for principal eigenvector similarity analysis. According to the presented results, it is possible to observe the high similarity between network traffic without attack, which was larger than 0.9610 for all evaluated cases, and emphasize the expressive low similarity when it was evaluated the traffic with the simulated port scan attack ($n = 15$), which was lower than 0.0276 for all evaluated approaches.

Table 2.4 Principal eigenvector similarity analysis for port scan detection

Time Frame q	**Time** n	**Similarity Analysis**			**Ground Truth**
		Incremental Individualized	**Incremental**	**Individual**	
3	1	0.9946	0.9946	0.9946	no
3	2	0.9934	0.9934	0.9999	no
3	3	0.9912	0.9912	0.9999	no
3	4	0.9888	0.9888	0.9999	no
3	5	0.9856	0.9856	0.9998	no
3	6	0.9840	0.9840	0.9999	no
3	7	0.9824	0.9824	1.0000	no
3	8	0.9794	0.9794	0.9999	no
3	9	0.9673	0.9673	0.9926	no
3	10	0.9674	0.9674	0.9997	no
3	11	0.9733	0.9733	0.9993	no
3	12	0.9702	0.9702	0.9993	no
3	13	0.9677	0.9677	0.9999	no
3	14	0.9646	0.9646	0.9998	no
3	15	0.0216	0.0216	0.0276	yes
3	16	0.9621	0.0209	1.0000	no
3	17	0.9611	0.0199	0.9998	no
3	18	0.9612	0.0191	0.9999	no
3	19	0.9613	0.0186	0.9998	no
3	20	0.9638	0.0190	1.0000	no

Comparing the approaches for similarity analysis, it is possible to observe that all approaches highlight the low similarity when evaluated the traffic under attack. However, the incremental approach figured out low similarity for times without attack, where $n = 16, 17, 18, 19, 20$, what indicates that the incremental approach can produce false positive results. This behavior occurs because the incremental approaches appends all selected traffic into the reference traffic for comparison against the original reference traffic, what makes more evident the first lack of similarity but reduces the changing detection capability after an attack detection.

Table 2.5 presents the results of the evaluation of the similarity analysis of principal eigenvectors for synflood detection. It shows the evaluation of the time frame $q = 4$, when the synflood attack is simulated, considering the incremental individualized, incremental and individual approaches for principal eigenvector similarity analysis. According to the results, it is possible to observe the high similarity between network traffic without attack, which is larger than 0.9907 for all evaluated cases, and emphasize the expressive low similarity when evaluated the traffic with synflood attack (between $n = 11$ and $n = 20$), which is lower than 0.1244 for all evaluated approaches.

Table 2.5 Principal eigenvector similarity analysis for synflood detection

Time Frame q	Time n	Similarity Analysis			Ground Truth
		Incremental Individualized	Incremental	Individual	
4	1	1.0000	1.0000	1.0000	no
4	2	0.9999	0.9999	1.0000	no
4	3	0.9997	0.9997	0.9999	no
4	4	0.9998	0.9998	1.0000	no
4	5	0.9965	0.9965	0.9908	no
4	6	0.9975	0.9975	1.0000	no
4	7	0.9977	0.9977	1.0000	no
4	8	0.9980	0.9980	1.0000	no
4	9	0.9987	0.9987	0.9999	no
4	10	0.9991	0.9991	1.0000	no
4	11	0.0085	0.0085	0.0284	yes
4	12	0.0162	0.0120	0.0343	yes
4	13	0.0248	0.0158	0.0427	yes
4	14	0.1243	0.0185	0.1041	yes
4	15	0.0082	0.0162	0.0103	yes
4	16	0.0404	0.0070	0.0580	yes
4	17	0.0397	0.0007	0.0573	yes
4	18	0.0408	0.0042	0.0584	yes
4	19	0.0408	0.0079	0.0584	yes
4	20	0.0477	0.0092	0.0757	yes

The incremental approach produces better results if compared with other evaluated approaches, with lower values and maximum of 0.0185 for times under attack, but this approach presents change detection limitation after the first outlier of similarity, in accordance to the results shown in Table 2.4 for port scan detection.

Comparing the incremental individualized and the individual approaches for principal eigenvector similarity analysis, it is possible to observe that the incremental individualized approach obtain lowest values for almost all cases, except for the time $n = 14$, where incremental individualized approach identified a larger similarity than the individual approach. The incremental individualized appends information about each evaluated traffic, therefore it incorporates traffic behaviors that can reduce the outlier capability detection, as occurred for the time $n = 14$.

Table 2.6 presents the results of the principal eigenvector similarity analysis evaluation for fraggle detection. For fraggle attack detection, the lack of similarity between legitimate and malicious traffic was more evident than for the evaluation of synflood and port scan detection. This behavior can be explained by the number of packets generated through the fraggle attack simulation, that was significantly larger than the number of packets generated during the synflood simulation. Considering the three approaches, the largest value for times under attack was 0.0083, while the shortest value for times without attacks was 0.9993.

Table 2.6 Principal eigenvector similarity analysis for fraggle detection

Time Frame q	Time n	Similarity Analysis			Ground Truth
		Incremental Individualized	Incremental	Individual	
5	1	1.0000	1.0000	1.0000	no
5	2	0.9999	0.9999	1.0000	no
5	3	1.0000	1.0000	1.0000	no
5	4	0.9999	0.9999	1.0000	no
5	5	0.9993	0.9993	0.9997	no
5	6	0.9993	0.9993	0.9997	no
5	7	0.9994	0.9994	1.0000	no
5	8	0.9995	0.9995	1.0000	no
5	9	0.9995	0.9995	1.0000	no
5	10	0.9995	0.9995	1.0000	no
5	11	0.0031	0.0031	0.0021	yes
5	12	0.0019	0.0025	0.0009	yes
5	13	0.0030	0.0026	0.0020	yes
5	14	0.0030	0.0027	0.0020	yes
5	15	0.0030	0.0028	0.0020	yes
5	16	0.0012	0.0025	0.0002	yes
5	17	0.0030	0.0026	0.0020	yes
5	18	0.0030	0.0026	0.0020	yes
5	19	0.0030	0.0027	0.0020	yes
5	20	0.0069	0.0023	0.0083	yes

Therefore, considering the evaluation for port scan, synflood and fraggle detection, the incremental approach can produce false positive results, while the individual and incremental individualized approaches produce quite similar results, even though the individual approach be more simple and require less memory and processing time.

These results highlight the capability of change detection based on similarity between legitimate and malicious traffic from flood or port scan attacks, by means of a subspace learned via EVD and MOS, endorsing the effectiveness and safety for adoption of threshold for attack detection through principal eigenvector similarity analysis.

Moreover, these results answer positively the question Q_1, which ask "Can the analysis of patterns from a learned subspace identify and detect anomalies in network traffic?". The results rejects the null hypothesis $H_1^{(N)}$ and confirms the alternative hypothesis $H_1^{(A)}$, which argues that "A subspace learned by eigenvalue decomposition can be used to detect and identify network attacks".

2.4.5.2 Port Analysis

Given $\hat{N}$, which is the set of estimated n-th minutes under attack, it is possible to apply cosine similarity analysis to identify variation between the principal eigenvectors, caused by the insertion of anomalous network traffic by a selected m-th port, during a n-th minute. Therefore, the incremental individualized and individual approaches of principal eigenvector similarity analysis were evaluated, for detection of ports under flood and port scan attacks, according to results presented in following tables. For this evaluation, the last principal eigenvector without attack $\boldsymbol{v}$ was used as reference for similarity analysis against each target port m-th.

Table 2.7 presents the results of the evaluation of principal eigenvector similarity analysis for detection of ports under port scan attack, showing only the time frame $q = 3$ and minute $n = 15$, due to the simulated port scan attack occurred only at this time, although the remain time frame has been completely evaluated and presented high similarity to the reference of traffic without network attack.

Table 2.7 Principal eigenvector similarity analysis for detection of ports under port scan attack (q=3 and n=15)

Port p	Approaches		Ground Truth
	Incremental Individualized	Individual	
80	0.9999	0.9999	no
443	0.9999	0.9999	no
53	0.9999	0.9999	no
21	0.9999	0.9997	yes
22	0.0298	0.9997	yes
23	0.0298	0.9997	yes
25	0.0298	0.9997	yes
110	0.0298	0.9997	yes
143	0.0298	0.9997	yes
161	0.0298	0.9997	yes
69	0.0298	0.9997	yes
123	0.0298	0.9997	yes
445	0.0298	0.9997	yes
600	0.9999	0.9999	no
19	0.9999	0.9999	no
67	0.9999	0.9999	no
68	0.9999	0.9999	no

The incremental individualized approach presented more sensibility to anomaly detection than the individual approach, the former produced the identification of a low similarity of 0.0298 for almost all ports under attack, unless the port 21, although the simulation has attacked this port. The individual approach was not able to identify low similarity for ports under attack, resulting in values of 0.9997 for ports with anomalous traffic and 0.9999 for ports without network attack.

For the evaluation of the proposed approaches for identification of ports under synflood and fraggle attack, all minutes of time frames under were analyzed. However, due to space limitations, only the results of the first minute, where a low similarity was identified, are shown. Nevertheless, the results obtained for the evaluation of traffic without attack presented high similarity to the reference traffic, with similarities close to 0.9999, and the evaluation of the other minutes under attack presented results quite similar to the results shown in the Tables 2.8 and 2.9.

Table 2.8 presents the results of the evaluation of principal eigenvector similarity analysis for detection of ports under synflood attack, showing only the time frame $q = 4$ and minute $n = 11$.

Table 2.8 Principal eigenvector similarity analysis for detection of ports under synflood attack (q=4 and n=11)

Port p	**Approaches**		**Ground Truth**
	Incremental Individualized	**Individual**	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	no
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	0.0077	0.0427	yes
19	1.0000	1.0000	no
67	1.0000	1.0000	no
68	1.0000	1.0000	no

According to results presented in Table 2.8, both approaches identifies low similarity for the traffic of port 600, which is the target port of the simulated synflood attack, but the incremental individualized approach identifies the lowest similarity and presents better sensibility to identification of synflood attack through principal eigenvector similarity analysis assisted by threshold definition.

Table 2.9 presents the results of the evaluation of principal eigenvector similarity analysis for detection of ports under fraggle attack, showing only the time frame $q = 5$

and minute $n = 11$.

Table 2.9 Principal eigenvector similarity analysis for detection of ports under fraggle attack (q=5 and t=11)

Port p	Approaches		Ground Truth
	Incremental Individualized	Individual	
80	1.0000	1.0000	no
443	1.0000	1.0000	no
53	1.0000	1.0000	no
21	1.0000	1.0000	no
22	1.0000	1.0000	no
23	1.0000	1.0000	no
25	1.0000	1.0000	no
110	1.0000	1.0000	no
143	1.0000	1.0000	no
161	1.0000	1.0000	no
69	1.0000	1.0000	no
123	1.0000	1.0000	no
445	1.0000	1.0000	no
600	1.0000	1.0000	no
19	0.0031	0.0004	yes
67	1.0000	1.0000	no
68	1.0000	1.0000	no

The results for the evaluation of ports under fraggle attack, shown in Table 2.9, were similar to the results obtained for synflood analysis, with the identification of low similarity for traffic of the port under attack. Nevertheless, for fraggle analysis, the individual approach identified the lowest similarity, that is 0.0004 while the incremental individualized approach obtained a similarity of 0.0031.

The incremental individualized approach was able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches presented false positives or low sensibility to principal eigenvector similarity analysis for network attack detection. This approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

According to the shown significant lack of similarity between legitimate and malicious traffic, it is possible to adopt safe thresholds for flood and port scan detection through principal eigenvector similarity analysis.

2.4.6 DARPA Scenario

This subsection presents a summarized view of results obtained from the application of the Eigensimilarity, focusing on the largest eigenvalue analysis, model order selection and the eigenvalue analysis, for flood and probe attack detection in the DARPA 1998 data set. Since the proposed framework is detailed in Section 2.3 and in Subsections 2.4.1, 2.4.2, 2.4.3, 2.4.4 and 2.4.5, here the focus is on the parameter selection, data set evaluation and results for flood and probe attack identification.

The DARPA data set includes 7 weeks of sniffed traffic saved into raw network packet data, i.e. pcap files. The traffic and the labeled attacks are grouped by week and day, with information of the number and types of attacks per day, but also providing the start time for each labeled attack. For this evaluation, an evaluation per day was performed, considering the network traffic of 24 hours split into Q time frames of 60 minute s ($N = 60$) and aggregate by minute and by port number. For each time frame q, a traffic matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{17\times20}$ is obtained, considering the ports 20, 21, 22, 23, 25, 79, 80, 88, 107, 109, 110, 113, 115, 143, 161, 389 and 443.

Since the proposed framework focus on flood and probe attack detection, only the attacks with behavior similar to flood or probe attack were evaluated. Initially all DoS and probe attacks were selected, but it was observed that the most cases of DoS focus on exploit system vulnerabilities instead of flooding attack, and most of probe attacks focus on ICMP instead of port scanning. Therefore, for evaluation of the proposed approach for flood and probe attack detection, it is necessary to select cases that implements flood or port scan behaviors. The following week-day-attack cases were selected:

1. week3-thursday-neptune;
2. week4-friday-portsweep;
3. week5-thursday-neptune;
4. week5-thursday-portsweep;
5. week5-friday-portsweep;
6. week6-wednesday-neptune;
7. week6-thursday-neptune;
8. week7-wednesday-portsweep.

Table 2.10 presents the evaluated results for attack detection, considering rates of TP, FP [64] and misclassification, which is defined as $\frac{(FN+FP)}{(TP+FP+FN+TN)}$ [19].

The analysis based on sample covariance of zero mean variables is evaluated for flooding behavior of netpune attacks, obtaining rates of 100.00 % for true positive (TP) detection and 60.00 % for false positive (FP) detection from 30 time frames. The results also show 50.00 % of misclassification rate, which attempts to estimate the probability of disagreement between the true and predicted cases by dividing the sum of FN and FP by the total number of pairs observed. The result for FP and misclassification analysis is poor due to the legitimate traffic of DARPA data set presents high number of packets per time from one source to one target, with no variation on IP source or target port. This observation corroborates with previous evaluations of the DARPA data set that highlight issues regarding traffic redundancy.

Table 2.10 Results of the attack detection evaluation

Solution	Attack Type	Metric	Result
Eigensimilarity	Flooding	True Positive	100.00 %
Eigensimilarity	Flooding	False Positive	60.00 %
Eigensimilarity	Flooding	Misclassification	50.00 %
Eigensimilarity	Probe	True Positive	76.92 %
Eigensimilarity	Probe	False Positive	18.52 %
Eigensimilarity	Probe	Misclassification	32.73 %
Callegari *et al.* [24]	Flooding	True Positive	82.00 %
Callegari *et al.* [24]	Flooding	False Positive	-
Callegari *et al.* [24]	Flooding	Misclassification	-
Lu and Ghorbani [25]	Overall	True Positive	94.67 %
Lu and Ghorbani [25]	Overall	False Positive	-
Lu and Ghorbani [25]	Overall	Misclassification	-
Lu and Ghorbani [25]	Portsweep	True Positive	50.00 %
Lu and Ghorbani [25]	Portsweep	False Positive	-
Lu and Ghorbani [25]	Portsweep	Misclassification	-

The analysis based on covariance of zero mean and unitary standard deviation variables was evaluated for port scan attacks, including probe attacks and DoS attacks that send few packets for several ports in order to exploit some vulnerability. The results show rates of 76.92 % for TP detection and 18.52 % for FP detection from 94 time frames. The observed misclassification rate for this scenario is 32.73 %. It was observed that all FN cases are probe attacks with a time delay between scanning one port and start scanning the next port, what can be called as sparse probe attacks. Cases with a delay of one minute or more were not detected by the proposed approach.

The performance of detection rate of flooding attacks is compared with the method proposed by Callegari *et al.* [24]. This work is a statistical method, based on PCA, without training or learning methods, even though it relies on visual analysis for principal components selection. The best detection rate of [24] was 82.00 % for detection of synthetically added flood attacks, while this current proposal obtains 100.00 % of detection rate for detection of flood attacks of DARPA data set. It is important to note that Callegari *et al.* [24] did not publish results of false positive and misclassification.

Due to the lack of statistical techniques without training or learning methods for detection of probe attacks, this proposed approach is compared to the Lu and Ghorbani's [25] proposal, which is a network anomaly detection model based on signal processing techniques that uses DARPA data set for evaluation. The results of [25] show the best detection rate of 94.67 % in terms of general attack instance detection, but shows a case case with 50.00 % of attack instance detection for the portsweep attack. The proposed approach presents 76.92 % of detection rate measured specifically for probe attacks, without the requirement of learning or training methods, in contrast to Lu and Ghorbani's [25] work.

2.5 Performance Evaluation

This section discusses the computational complexity and the performance evaluation of the proposed framework, focusing on the main steps, which are the eigenvalues decomposition (EVD), largest eigenvalues analysis, application of MOS scheme and principal eigenvector similarity analysis, according to Figure 2.7 and equations presented in Section 2.3.

2.5.1 Complexity Analysis

The EVD, calculated according to (2.6), requires the previous calculation of covariance matrix, according to (2.2), (2.3), (2.4) and (2.5). The covariance matrix calculation is $O(M^2N)$ and the EVD is $O(N^3)$, where M denotes the number of network ports and N denotes the period time. Therefore, the computational complexity for all steps for EVD can be represented as $O(M^2N+N^3)$ and yields an $O(N^3)$ upper bound on the worst-case running time for EVD.

EDC and EFT are the MOS schemes that presented accuracy on the evaluation for the network attack detection. The computational complexity evaluation for MOS focuses on EDC scheme, since EDC requires less processing time than EFT but presents the same accuracy for the evaluated scenario. EDC scheme is $O(Q\log Q+Q+Q\log Q)$ and its worst-case running time can be represented as $O(Q \log Q)$, where Q denotes the number of time frames.

The largest eigenvalue analysis is $O(\hat{d}Q)$, where $\hat{d}$ denotes the number of time frame under attack, according to Algorithm 1. Subsequently, the principal eigenvector similarity analysis relies on EVD and cosine similarity analysis, which is $O(N^2)$, for $\hat{d}$ time frames, therefore the principal eigenvector similarity analysis is which is $O(\hat{d}(M^2N + N^3 + N^2))$ and yields an $O(N^3)$ upper bound on the worst-case running time for principal eigenvector similarity analysis.

Therefore, the proposed framework is $O(N^3 + Q \log Q + \hat{d}Q + N^3)$ and its worst-case running time is $O(N^3)$. The computational complexity of EVD is predominant in the framework, but the approach splits the data into time frames with period time N, which makes possible to limit the growth of N even for evaluations of cases with total time larger than N, reducing the impact caused by the computational complexity of EVD.

2.5.2 Processing Time Analysis

For better understanding the scalability and impact of configurations of N, M and Q, the processing time required for different scenarios of parameter configurations and data set were evaluated, measuring the processing time of:

1. Eigen analysis based on sample covariance of zero mean;
2. Eigen analysis based on sample covariance of zero mean and unitary standard deviation;
3. EDC MOS scheme;

The performance evaluation focus on the main steps, which are discussed in subsection 2.5.1 regarding the complexity analysis. The data modeling is also a time consuming step, however its processing can be optimized through distributed processing techniques, such as MapReduce, achieving high throughput for packet counting or even for deep packet inspection [65, 66, 67, 68]. It is also possible to evaluate the adoption of faster SVD algorithms, considering implementations based on truncated or randomized approaches [69] that aim reduce the complexity and processing time.

The experiments were performed in a desktop computer with an Intel Core i7-4510U 2.00GHz and 16 GB of RAM, considering: variations on the network traffic time; the frame size denoted as N; the number of network ports denoted as M; the mean processing time for eigen analysis based on sample covariance of zero mean, denoted as 1-time; the mean processing time for eigen analysis based on sample covariance of zero mean and unitary standard deviation, denoted as 2-time; and the mean processing time for EDC MOS scheme, denoted as 3-time. The mean time calculations was obtained from 200 measurement repetitions, in order to obtain reliable values.

Table 2.11 presents the measured results. The experiment considered traffic time of 16, 20 and 22 hours, according to the selected traffic time per day available by the DARPA data set. Note that the processing time increases according to the increment in traffic time, around 2 or 3 times for 1-time and 2-time, but the worst measured processing time is 4.7250 milliseconds.

Table 2.11 Processing time of the main steps for anomaly detection

Traffic Time (hour)	Frame Size (min)	Num. Ports	1-time (ms)	2-time (ms)	3-time (ms)
16	10	17	0.7900	0.8100	0.0650
16	20	17	0.5250	0.5950	0.0100
16	60	17	0.9700	1.1400	0.0250
16	120	17	0.6050	0.6100	0.0050
16	60	34	1.2750	1.2200	0.0050
16	120	34	1.1200	1.1700	0.0050
20	10	17	2.7950	2.8950	1.1000
20	20	17	2.0700	2.0200	0.3500
20	60	17	1.0250	1.0450	0.0650
20	120	17	1.0000	1.0700	0.0350
20	60	34	2.9650	3.2100	0.0400
20	120	34	2.9950	3.1150	0.0200
22	10	17	4.7250	4.0850	1.4600
22	20	17	2.3200	2.6800	0.2450
22	60	17	1.0700	1.1200	0.0300
22	120	17	0.9900	1.0500	0.0250
22	60	34	3.0850	3.1250	0.0650
22	120	34	2.8100	2.9600	0.0250

According to Table 2.11, the processing time increases with the frame size *N* decreasing, therefore it is possible to evaluate the frame size that produces better identification rates and acceptable processing time. The number of ports evaluated during the proposed scheme is also an important variable regarding processing time optimizations, since the significant increase of processing time observed between scenarios considering 17 or 34 ports, with growth between 7% and 199%.

2.6 Conclusion and Future Works

This work models the network traffic as a signal processing formulation for applying to the framework for detection and identification of network attacks, named Eigensimilarity, which is based on subspace decomposition, eigenvalue analysis, model order selection (MOS) and principal eigenvector similarity analysis.

The Eigensimilarity is evaluated and the experimental results show that synflood, fraggle and port scan attacks can be detected accurately and with great detail in an automatic and blind fashion, without training, applying signal processing concepts for traffic modeling and through approaches based on MOS and principal eigenvector similarity analysis from a subspace obtained by EVD. Therefore, the observed results rejects the null hypothesis $H_1^{(N)}$ and confirms the alternative hypothesis $H_1^{(a)}$.

The main contributions of this work were: the extension of an approach based on MOS combined with eigen analysis to blindly detect time frames under network attack; the proposal and evaluation of an principal eigenvector similarity based framework to identify details of network attacks, presenting accuracy of timely detection and identification of TCP/UDP ports under attack, as well as presenting acceptable complexity and performance regarding the processing time.

The incremental individualized approach of principal eigenvector similarity analysis, is able to detect low similarity for all evaluated scenarios and types of network attack, while the other approaches present false positives or low sensibility to principal eigenvector similarity analysis for network attack detection. Therefore, the incremental individualized approach is able to gradually and incrementally adapt to network traffic changing, preserving the sensibility to identify outliers or anomalies by time or network port, and reducing the occurrence of false positives.

The principal eigenvector similarity analysis is applied for each dimension individually, for the time dimension initially and after for port dimension, in order do identify lack of similarity. Future work can be directed to pairwise similarity analysis, where the two dimensions are not evaluated separately, or for a principal component analysis with indication of element-wise anomalies, such as the Robust Principal Component Analysis (RPCA).

Considering the reported limitations for the DARPA and KDD data sets, and the lack of available labeled data sets for network attack detection, future work can be directed to evaluate our proposed framework for novel and up-to-date data sets of network attack detection.

Future research can be directed to improvements for better false positive rates, as well as to make the proposed framework able to identify sparse probe attacks or subtle behaviors, such as exfiltration or covert communication, considering the evaluation of a flow-based analysis and novel data sets. Distributed or parallel processing can also be evaluated to analyze the scalability and processing capacity for monitoring high throughput network traffic, as well as it is possible to evaluate approaches with less complexity for EVD [69]. Future research can also evaluate the application of the proposed approach to different attack types and domains, considering cases that are aware to behavioral analysis.

We adopted a data modeling based on two dimensions, which are the time and the network ports. Future works can evaluate improvements given by a multidimensional modeling and adoption of tensor-based approaches [70, 71], in order to evaluate complex patterns that can be revealed by multidimensional analysis and tensor-based decomposition. It is possible to consider the addition of a seasonal dimension for our proposed data modeling, in order to evaluate the the relationship between the selected season, such as week or month, to the analysis of amount of packets per network port and time.

3

Eigensimilarity for Anomaly Detection in Offline Mobile Client

The protection scheme used in a mobile device should be both computationally secure as well as resource-constrained due to battery power limitations [72]. Therefore, encrypting files and generating keys on a mobile device is not considered a good solution. On the other hand, the protection schemes with good computational qualities lack the security analysis in many cases [73]. The common practice is the passive monitoring of user activities by online agents [74]. However, the mobile device usage stays unprotected in all the proposed scenarios while in offline mode. When the mobile client goes offline with the sensitive corporate data on board all powerful cloud-based tools can not help and the mobile client has to secure itself with its own limited resources [75]. Moreover, due to the resources constraint, there is a crucial difference in strategy of online and offline mode protection.

Additional security issues and requirements have to be considered when mobile clients are actively used in corporate cloud environment [74]. Today more and more organizations and enterprises are functioning in the Bring-Your-Own-Device (BYOD) paradigm. The uncontrolled usage of the mobile devices represents a serious risk to the development of secure SME cloud platforms being the bottleneck of the corporate information security system (ISS). While the enterprise cloud infrastructure based on the web interface can be protected by powerful third-party services, such as CASB and CAC, the corporate mobile client is usually light-weighted and generally less protected.

In this chapter we propose an architecture and approach for user behavior analysis based on Eigensimilarity [1], in order to detect possible threats in offline corporate mobile applications [6, 3]. The key expiration period is safely incorporated into the proposed system solution in order to enhance security and the behavioral analysis can indicate malicious behaviors, their variations, as well as novel attacks, which present low or high variance in comparison to legitimate user behaviors.

The work is structured as follows. In Section 3.1 we analyze the most common security problems in the mobile cloud environment and the solutions for offline protection in the BYOD world. The mobile security architecture of the proposed solution is outlined in Section 3.2, and in Section 3.3 is presented the detailed scheme of the proposed solution to the problem of offline mobile client security. In Section 3.4 we explain the use of Eigensimilarity and in Section 3.5 we discuss the common threat scenarios, the data modeling, the performance analysis and discussion of the practical implementation results. Finally, in Section 3.6 the chapter is concluded.

3.1 Related works

The increasing usage of BYOD demands more sophisticated data protection services compared to ordinary computing environment. A common practice is to provide additional contextual methods apart from authentication, DLP services, and encryption, which can be at rest, in transit and in use [74, 72, 76, 77]. The contextual methods increase the security of the mobile client at a maximum level with minimum resource requirements. The most commonly used are:

1. Using geolocation of the device to trace its usage;
2. Setting up the expiration period of an app;
3. Setting up the expiration period of client pass pin;
4. Setting up the counter of failed tries;
5. Secured transfer politics between apps;
6. Restricting access to the corporate app;
7. Restricted or prohibited offline access;
8. Logging and auditing.

Preventing data leakage on the mobile device is a crucial security problem. Therefore, it is necessary to take additional control and protection measures for the confidential data on the mobile devices that leave the boundaries of the organization. Generally, the most sensitive and confidential data should not be permitted to be transferred to the mobile device. However, what if the SME needs to allow their employees to work on such devices and even use them on the offline mode for the convenience and traffic reducing, or even for a particular characteristic of the mobile client or the business itself?

From the theoretical point of view of this problem, there are several surveys, whose common point is the mobile cloud as a rapidly developing paradigm that poses many security and complexity problems [77, 74, 76, 72, 78, 79, 80, 81, 82, 83]. Kulkarni and Khanai [81] discuss the most important threats related to Mobile Cloud Computing and argue that there is a need for a lightweight secure framework that provides security with minimum communication and processing overhead on mobile devices.

An analysis of the new models of mobile cloud computing and new ways of using data storage services is presented in [72, 77, 78, 79]. Commonly, the models and protection schemes concentrate on the encryption properties and either perform the computations on their own [84, 85] or use the cloud provider to off-load the cryptographic operations [86, 87]. Obviously, it is natural the mobile client cannot handle all operations securely without the assistance of a cloud provider, due to resources constraint and battery limitation.

The necessity to use schemes that function without putting load on a provider arises when it is desired to make the mobile client less dependent on the cloud provider, i.e. corporate client continues to provide the secure access to the sensitive data without connection to the network. As discussed in [72], all the currently known schemes of encryption, performing the computation, either use a cloud provider [84], a third party trusted agent [85, 83], a combination of both [86] or *ad hoc* approaches [82]. In some cases, they concentrate on computational complexity without taking care of user privacy and security [87]. Therefore, according to [77, 72], the state-of-the-art mobile cloud security models do not consider the problem of the offline security mode.

To the best of our knowledge, the offline mode security problem has not been deployed yet, neither in academia nor in the industry [74, 72, 76, 77]. Therefore, the main concern of this proposal is the protection of the mobile client and its data in offline mode, when the functions of data protection cannot be offloaded to a cloud or a trusted party.

3.2 The mobile security architecture

The approach proposed in this work describes and implements a mobile client security infrastructure for malicious behavior analysis. The mobile security processes depend on the key expiry period, and are used to access the protected storage. Once the user keys expire, the user is requested to enter his valid credentials, i.e. PIN and password. The mobile client then sends the credentials to the server for verification. Once the new set of access keys is received, the user can access the protected files in the offline mode, without the access to the server. This means that no further communication with the server is needed until the key expires.

The core set of functions and protocols of the proposed architecture can be divided into three sets of operations as shown in Figure 3.1.

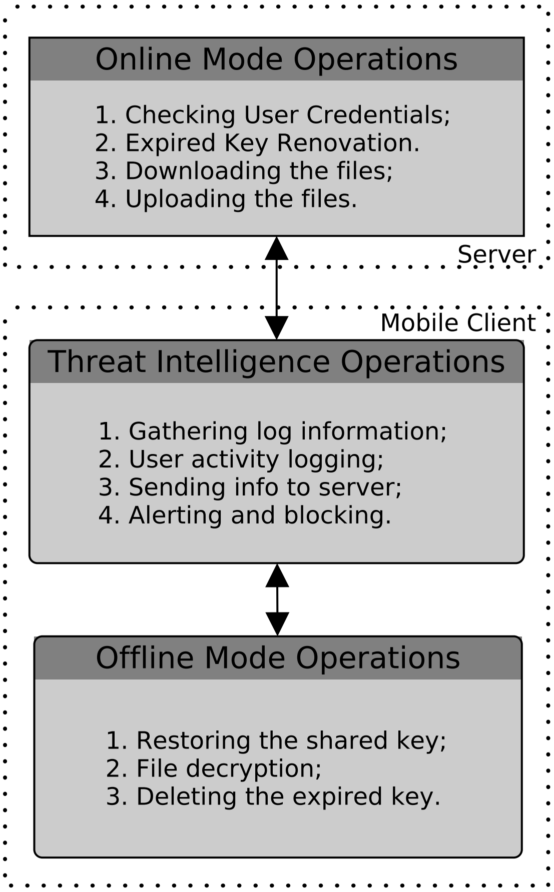

Online Mode Operations
1. Checking User Credentials;
2. Expired Key Renovation.
3. Downloading the files;
4. Uploading the files.
Server
Mobile Client
Threat Intelligence Operations
1. Gathering log information;
2. User activity logging;
3. Sending info to server;
4. Alerting and blocking.
Offline Mode Operations
1. Restoring the shared key;
2. File decryption;
3. Deleting the expired key.

Figure 3.1 The core set of functions and protocols of the mobile cloud security infrastructure

Figure 3.1 describes the mobile client protection both in online and offline mode. In online mode, the client has the possibility to connect to the server and the security of the client is enhanced by the server-backed up mechanisms. On the other hand, in offline mode the client's security is supported by the standalone mechanisms. Additionally, the mobile client protection is enhanced by the threat intelligence unit providing the constant monitoring and analysis.

Figure 3.2 depicts the client-side protection mechanisms. The client should support 4 subsystems:

1. Encryption subsystem that provides the procedures of encryption and decryption;

2. Protected storage subsystem that provides the downloaded shares and key storage protection;

3. The Threat intelligence unit that provides the constant monitoring;

4. The communication subsystem that enables with the server.

In summary, all security procedures are connected to 4 groups of operations: file request and receiving; encryption and decryption; file and key storing; monitoring and analysis.

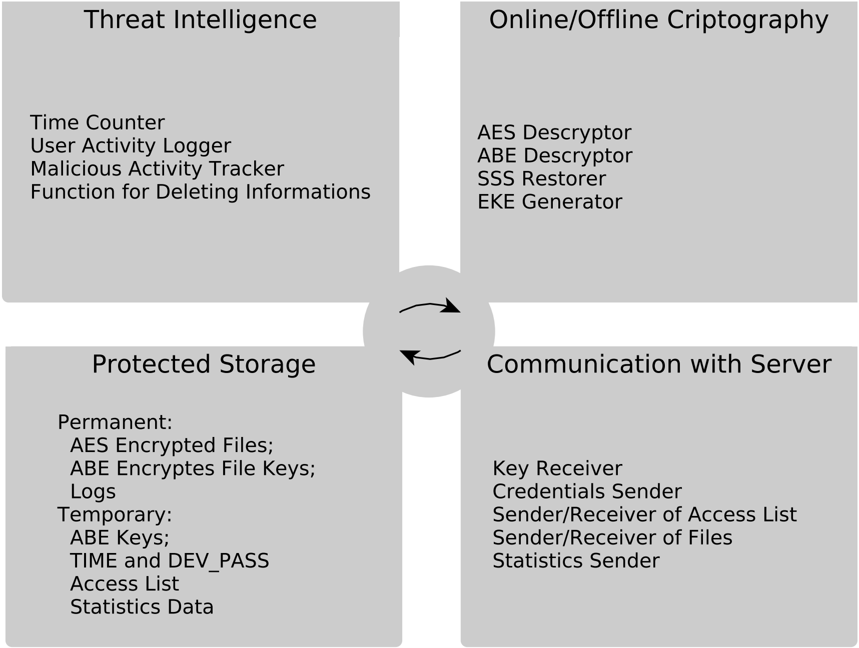


Figure 3.2 The Client-side Architecture

This architecture consists of the modules of cryptographic functions, threat intelligence infrastructure, communication with server and storage. However, this work focus on the threat intelligence infrastructure module for malicious behavior analysis.

The threat intelligence infrastructure takes into account simple actors such as the time counter for the key expiry period, the counter of unsuccessful tries in order to protect from brute force attacks, and Eigensimilarity-inspired statistics analyzer. Functions such as alerting and deleting the expired key belong to this module as well. These functions are described in Subsection 3.4.

3.3 The proposed solution for offline mobile security

This section proposes an approach for the mobile client protection in which the security is supported in offline modes. Currently and to the best of our knowledge, the systems of mobile client protection follow a model where the protected client can operate only when it is connected to the cloud, which is not always convenient for the end-user. The basic principles of the mobile client protection herein proposed are:

1. Optimized communication with the cloud when the mobile client does not need to be constantly connected to the server due to the resource constraint and necessity to secure this communication;

2. Optimized combination of the security mechanisms so that the mobile client does not need to perform complex computation like encryption and key generation due to its resource constraint;

3. Behavioral analysis of user's operations on mobile client, which can indicate anomalous or automated activities performed by attackers.

The most important security issues in the proposed model arise when the client goes to the offline mode and the user is still allowed to get the access to the protected SME documents. In this case, the server can neither monitor the user activity nor provide the protection methods. The security should be performed at the mobile client. Additionally, the maximum protection should be provided at the minimum resource cost.

In the online mode the mobile client uses the secure communication with the server in order to verify the validity of user's credentials. On the contrary, the offline protection model should be approached independently.

The Figure 3.3 describes the proposed architecture for offline mobile security, showing the modules responsible for securing the mobile client, which includes:

1. **Protected Storage**: the storage is protected with the shared user key and contains the ABE keys giving access to the file keys which allow decrypting the stored files.

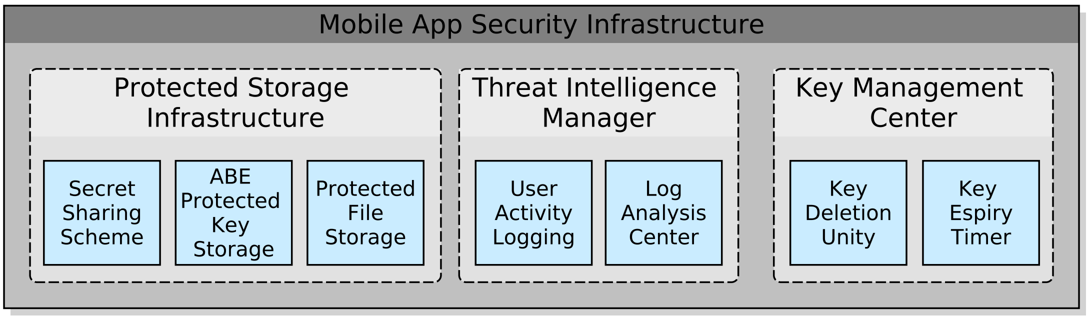


Figure 3.3 Proposed Architecture for Offline Mobile Security

2. **Threat Intelligence Manager (TIM)**: most attacks incur into significant variation on the legitimate behavior of information systems, or they adopt well-known patterns that can be easily detected by monitoring the system in the case of the offline mode. Signal processing techniques have been successfully applied to anomaly detection [88] and have become a solution to a problem of improving detection accuracy, adaptability and computational cost for application on resource-constrained scenarios. Therefore, signal processing can be applied in offline mobile client security, for evaluating anomalies on user's behavior, according to the scenarios in Section 3.5. Moreover, the Eigensimilarity, which is an approach based on subspace learning and on effective signal processing technique to separate noise components from the principal components [89, 71], named Model Order Selection (MOS), can be applied into anomaly and attack detection [1], to identify and separate malicious behaviors from the legitimate ones. The TIM is an internal module of the mobile client that implements offline anomaly detection through signal processing techniques.

3. **Key Management Center**: it includes the functions for maintaining the key expiry period and deleting the expired keys.

3.3.1 Offline Behavioral Analysis

In the proposed client security architecture, the Threat Intelligence Manager (TIM) is responsible for receiving logged user operations, perform feature extraction, data modeling and malicious behavior analysis in order to identify possible threats, in offline mode. Figure 3.4 depicts the TIM workflow for offline behavioral analysis.

As depicted in Figure 3.4, users request operations are logged so that the main features

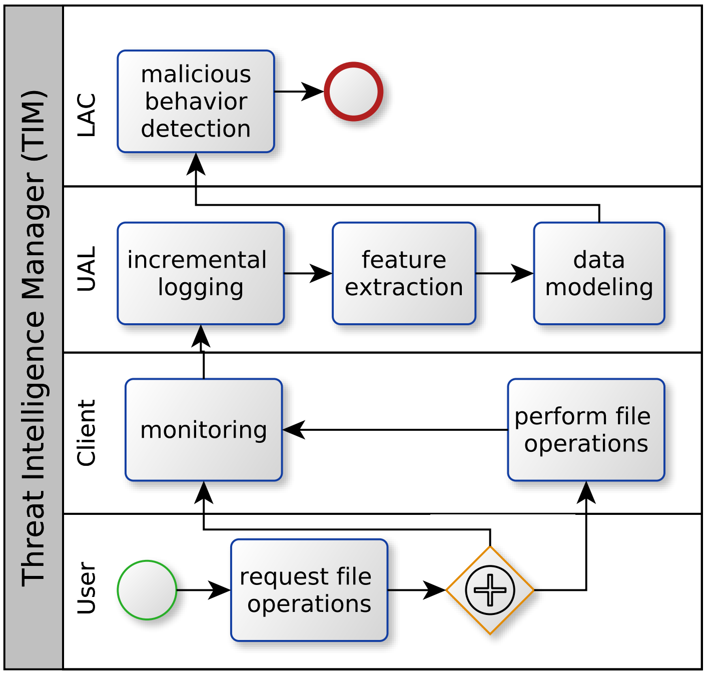


Figure 3.4 The Threat Intelligence Manager Workflow

can be monitored in the mobile client. The user behavior, trying or effectively executing operations, shall be incrementally captured and logged, making possible to monitor the main features that can reveal malicious behaviors, as well as to identify unexpected behaviors that can reveal possible threats. Therefore, the user operations are monitored by the client app, which sends the information to the User Activity Logging (UAL).

The UAL is responsible for the incremental logging of activities of the mobile client, feature extraction and data modeling for malicious behavior detection, through the Log Analysis Center (LAC). As an internal module of the mobile client, the UAL implements monitors of selected events of the application, such as a login attempt or a file decryption, and logs the desired information for further analysis. The logged information shall be decomposed into selected features and modeled as matrices, composed of the number of occurrences of the selected features by its location and by time. The resultant data is submitted to the LAC, for anomaly detection.

The LAC performs the behavioral analysis through eigenvalue analysis and MOS schemes proposed by Eigensimilarity, which identify anomalies on sparse, subtle or abrupt number of user operations. The malicious behavior detection is detail described in 3.4.

3.4 Eigensimilarity for Threat Intelligence

In the context of anomaly-based schemes for attack detection, the proposed behavioral analysis approach applies signal processing techniques, such as subspace learning by eigenvalue decomposition and Model Order Selection schemes [35], for automatic detection of attacks or malicious behaviors. Model Order Selection is an effective signal processing technique for several applications, allowing separating the only noise components from the principal components applying a rank reduction of the data.

Applying MOS to the analysis of user operations can be effective in order to reveal the occurrence of malicious behavior during an offline session. MOS for threat intelligence requires that the target features, such as user operations, should be modeled as a matrix composed by the number of occurrences grouped by location and time, and split into Q time frames. Therefore, the framework considers the time variations of the matrix $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$, with $q = 1, \ldots, Q$, in order to detect the occurrence of malicious behaviors. For example, one element of $\boldsymbol{X}^{(q)}$ can represent the number of file readings on folder m during the minute n, from file operations logged by the mobile client.

The Eigensimilarity can rely on sample covariance of zero mean variables (called as **zero mean covariance** for the sake of simplicity) and sample covariance of zero mean and unitary standard deviation (called as **zero mean and standardized covariance** for the sake of simplicity) variables, where the former is useful to identify abnormalities caused by large amounts of operations during a period, while the latter is applied to identify anomalies on sparse or subtle number of file operations.

Classical approaches to model order selection require the computation of the sample covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ and of its eigenvalues, obtained from the measurement matrix $\boldsymbol{X}$ of the zero mean samples given by

$$\boldsymbol{y}_m^{(q)} = \boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}. \tag{3.1}$$

The set of obtained vectors $\boldsymbol{y}_m^{(q)}$ composes the zero mean matrix $\boldsymbol{Y}^{(q)}$, then the zero mean covariance matrix $\hat{\boldsymbol{R}}_{yy}^{(q)}$ can be estimated as follows

$$\hat{\boldsymbol{R}}_{yy}^{(q)} = \frac{1}{N} \boldsymbol{Y}^{(q)} \boldsymbol{Y}^{(q)^{\mathrm{T}}}, \tag{3.2}$$

where $\hat{\boldsymbol{R}}_{yy}^{(q)}$ means the estimation of the sample covariance matrix from the measured zero mean matrix $\boldsymbol{Y}^{(q)}$ over N minutes of the time frame q.

For Eigensimilarity based on sample covariance of zero mean and unitary standard deviation, in order to identify anomalies with sparse or subtle behavior, it is required, for each variable, to make the standard deviation unitary as follows

$$\boldsymbol{z}_m^{(q)} = \frac{\boldsymbol{x}_m^{(q)} - \bar{\boldsymbol{x}}_m^{(q)}}{\boldsymbol{\sigma}_m^{(q)}}. \tag{3.3}$$

The set of vectors $\boldsymbol{z}_m^{(q)}$ composes the matrix $\boldsymbol{Z}^{(q)}$, then the zero mean and standardized covariance matrix $\hat{\boldsymbol{R}}_{zz}^{(q)}$ can be calculated via

$$\hat{\boldsymbol{R}}_{zz}^{(q)} = \frac{1}{N}\boldsymbol{Z}^{(q)}\boldsymbol{Z}^{(q)^{\mathrm{T}}}. \tag{3.4}$$

Once the $\hat{\boldsymbol{R}}_{yy}$ or $\hat{\boldsymbol{R}}_{zz}$ have been obtained for anomaly detection based on Eigensimilarity, for the sake of simplicity, we refer to $\hat{\boldsymbol{R}}_{yy}$ or $\hat{\boldsymbol{R}}_{zz}$ as a matrix $\hat{\boldsymbol{R}}$. Therefore, the next step of the algorithm is the eigenvalue decomposition (EVD), calculated according to $\hat{\boldsymbol{R}}^{(q)} = \boldsymbol{V}^{(q)}\boldsymbol{\lambda}^{(q)}\boldsymbol{V}^{(q)^{\mathrm{T}}}$, in order to obtain the vector of eigenvalues $\boldsymbol{e}$, as following:

$$\boldsymbol{e}^{(q)} = \mathrm{diag}(\boldsymbol{\lambda}^{(\mathrm{q})}), \tag{3.5}$$

The eigenvalues should be sorted in descending order, as defined by $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_m^{(q)}$, to make possible the selection of the first eigenvalue in the obtained sequence, represented by $\lambda_1^{(q)}$, which is the largest eigenvalue of the data evaluated for attack detection.

The process of obtaining the $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$ and the matrix $\hat{\boldsymbol{R}}^{(q)}$, finding the largest eigenvalue for each q-th time frame, should be repeated until $q = Q$, in order to obtain the largest eigenvalue of all time frames, as presented by

$$\boldsymbol{E} = \begin{bmatrix} \lambda_1^{(1)} & \lambda_1^{(2)} & \lambda_1^{(3)} & \cdots & \lambda_1^{(Q)} \\ \lambda_2^{(1)} & \lambda_2^{(2)} & \lambda_2^{(3)} & \cdots & \lambda_2^{(Q)} \\ \lambda_3^{(1)} & \lambda_3^{(2)} & \lambda_3^{(3)} & \cdots & \lambda_3^{(Q)} \\ \vdots & \vdots & \ddots & \vdots & \\ \lambda_m^{(1)} & \lambda_m^{(2)} & \lambda_m^{(3)} & \cdots & \lambda_m^{(Q)} \end{bmatrix}. \tag{3.6}$$

Since $\lambda_1^{(q)} > \lambda_2^{(q)} > \lambda_3^{(q)} > \cdots > \lambda_{m-1}^{(q)} > \lambda_m^{(q)}$, then the first line of the matrix $\boldsymbol{E}$ contains the largest eigenvalues of each q-th time frame, which is the expected input for MOS schemes and can be expressed as

$$\boldsymbol{e}_{\max} = [\lambda_1^{(1)}, \lambda_1^{(2)} \cdots \lambda_1^{(Q)}] \tag{3.7}$$

Once obtained the largest eigenvalues of each q-th time frame, it is possible to apply a selected MOS scheme to estimate the model order $\hat{d}$, which is the estimated number of time frames with malicious behavior. Therefore, $\boldsymbol{e}_{\max}$ is used as input parameter for MOS schemes, according to the equation

$$\hat{d} = \mathrm{MOS}(\boldsymbol{e}_{\max}) \tag{3.8}$$

Note that some MOS schemes may also require the amount of time that compose a time frame, such as $\hat{d} = \mathrm{MOS}(\boldsymbol{e}_{\max}, M)$. For more information about MOS schemes, interested readers are referred to section 2.3.2 and to [54].

3.5 Results and analysis

This section provides the detailed analysis of the results from the proposed approach regarding security and performance analysis.

3.5.1 Security analysis

The security analysis of the proposed model was performed from the user behavioral analysis. Two common attack scenarios were analyzed. First, the malicious outsider trying to infect or steal the important files. Second, the malicious expired user trying to steal the important files.

3.5.1.1 Common threat scenarios

This section provides the detailed description of the common scenarios in which the log and behavioral analysis is provided. The behavioral analysis can help to keep the user or administrator informed of the threat and take actions, as well as it can be useful in order to implement threat prevention or reactive actions to avoid threat propagation.

Scenario 1. *An attacker uses a valid password to perform operations on a bulk of files.*

The session time defines the period when operations can be performed until the next session renewing. During this period, it is still necessary to identify attacks and malicious behavior on file operations, in order to avoid fast attacks to perform unauthorized access to information or data modification. Some attacks present behavioral patterns based on abrupt number of operations, such as the ransomware attack, which is a growing attack [90] that blocks the access to valuable resources and requires a payment in order to unblock the content. The access to the resources can be blocked by the attacker through some techniques, when the content is encrypted by the attacker, the ransomware attack can be called cryptoransomware or cryptolocker [91].

The Eigensimilarity and MOS schemes based on zero mean covariance analysis are effective to reveal abrupt changing of behaviors over time [35], making possible to identify intense malicious behaviors on offline mode of mobile clients, such in case of ransomware attack or bulk access to sensitive data.

The large number of operations over time is a well-known pattern of some attacks, due to the efforts on security measures to make the attacks infeasible over time. In this context, the operations can also be evaluated in contrast to the estimated required time for operations done by legitimate behaviors, such as the evaluation of the mean time between operations, highlighting the occurrence of infeasible behaviors in comparison to legitimate user activities.

Sparse or subtle file operations, with low number of operations distributed over different files or directories, during short period of time can indicate anomalies in contrast to the required time for legitimate directory navigation. Eigensimilarity and zero mean and standardized covariance analysis can be suitable if applied to evaluate the time and location of operations, in order to identify unreachable navigation, if compared to legitimate navigation

The Eigensimilarity based on zero mean covariance analysis indicates abnormalities caused by large amounts of operations during a period. Subsequently, the eigenvalue analysis highlights massive or concentrated operations over time or folder location, which is evaluated by MOS schemes in order to identify the number of malicious behaviors during the evaluated time.

This threat scenario, where an attacker uses a valid password and session to perform operations on a bulk of files, can have its steps described as:

(a) The hacker has access to the mobile client and is able to perform operations;

(b) The session time is valid;

(c) The hacker tries to perform legitimate operations, such as file decryption, encryption, reading, writing or directory navigation;

(d) The mobile client incrementally append each operation attempt time into the logging;

(e) The TIM module evaluates the logging of legitimate operations, applying zero mean and standardized covariance analysis to identify anomalies on sparse or subtle number of file operations, highlighting the occurrence of infeasible behaviors in comparison to legitimate user activities;

(f) The TIM module evaluates the logging of legitimate operations, applying zero mean covariance analysis to identify abnormalities caused by massive operations during the session time.

Scenario 2. *Usage of expired password to perform unauthorized operations.*

In the offline mode, the session time is used to restrict the operations during a specified period, although it is possible to manipulate the current time in mobile device, to emulate a period in which the session was valid. The log analysis by Eigensimilarity can deal with this kind of threat, through the incremental logging of the time when each operation was performed, followed by the behavioral evaluation of operations over time.

The incremental logging assumes that new logged operations shall have equal or bigger time than the last logged operation, the violation of this rule means that the system is out of sync and indicates a malicious behavior. Additionally, a large amount or sparse operation performed at the same time, or during a short period, can indicate the use of backtrack techniques to maintain a valid session during necessary time to perform an attack. Massive, subtle or sparse malicious operation performed during a valid session time can be identified by MOS schemes based on covariance analysis.

Applying Eigensimilarity to the analysis of the time between user operations can be effective in order to reveal the occurrence of malicious behavior during an offline session. The MOS based on zero mean and standardized covariance analysis identifies anomalies on sparse or subtle variation in the number of file operations, since the eigenvalue analysis highlights the unexpected number of sparse (such as file operations on diverse folders) or subtle operations. Consequently, the result of the eigenvalue analysis is applied to MOS schemes, in order to identify the occurrence of malicious behaviors during the valid session.

The Eigensimilarity based on zero mean covariance analysis indicates abnormalities caused by large amounts of operations during a period. The eigenvalue analysis based on the zero mean covariance matrix highlights massive or concentrated operations over time or location, which is evaluated by MOS schemes in order to identify the number of malicious behaviors during the evaluated time.

This threat scenario, where the attacker uses expired password to perform unauthorized operations, can have its steps described as:

(a) The hacker steals the operating system;

(b) The hacker modifies the time of the operating system to a period when the session was valid;

(c) The hacker has access to the mobile client and is able to perform operations;

(d) The hacker tries to perform legitimate operations, such as file decryption, encryption, reading, writing or directory navigations;

(e) The mobile client incrementally append each operation attempt time into the logging;

(f) The mobile client verifies if one logged time is older than the last operation time. If it is true, the Eigensimilarity module classifies the evaluated operation as malicious;

(g) The TIM module evaluates the logging of legitimate operations, applying zero mean and standardized covariance analysis to identify anomalies on sparse or subtle number of file operations;

(h) The TIM module evaluates the logging of legitimate operations, applying zero mean covariance analysis to identify abnormalities caused by massive operations during the session time;

3.5.1.2 Data Modeling for Behavioral Analysis

The Eigensimilarity and MOS schemes are used in order to identify anomalous behavior that can indicate an attack and be used to prevent or avoid attack propagation. Therefore, it is necessary to analyze the data that can be collected from user operations on mobile client, to identify features that can be modeled and submitted to Eigensimilarity, according to described in Section 3.4.

Thus, the data is grouped into time frames $\boldsymbol{X}^{(q)} \in \mathbb{R}^{M \times N}$, with $q = 1, 2, 3, \ldots, Q$, where M defines the decomposition of a selected feature, N defines the time decomposition and represents the number of occurrences of the feature m during the time n.

In offline mode, the user is still allowed to get access to operations that do not require communication with the server side. These operations and their selected features shall be incrementally logged by the UAL of the mobile client, in order to be analyzed by the TIM to identify malicious behaviors.

This work proposes to evaluate the following features, which represents events of the user operating the mobile client.

3.5.1.2.1 File Access (Time and File System Location), i.e. data access to selected files in offline mode, accessing the data stored on the mobile client. The file access feature can be decomposed into more detailed features, which are:

1. number of file decryption;
2. number of decrypted file reading;
3. number of decrypted file execution.

Therefore, it is necessary to generate three matrices for the following malicious behaviors analysis:

(a) massive file access, which can reveal data leakage and be identified by MOS schemes based on zero mean covariance analysis;

(b) low file access into several folders, characterized by sparse operations that can reveal unreachable navigation performed by automated file accesses in order to avoid the massive file access characterization;

(c) Malicious sparse file accesses can be identified by MOS schemes based on zero mean and standardized covariance analysis.

3.5.1.2.2 File Update (Time and File System Location), i.e. writing operations into selected files in offline mode, writing the data stored on the mobile client. The update feature can be decomposed into:

1. number of file encryption;
2. number of decrypted file writing.

Therefore, it is necessary to generate two matrices for malicious behaviors analysis, such as:

(a) massive file update, which can reveal ransomware or similar attacks and be identified by MOS schemes based on zero mean covariance analysis;

(b) low number of file update into several folders, characterized by sparse operations that can reveal unreachable navigation performed by automated file accesses in order to avoid the massive file access characterization. Malicious sparse file accesses can be identified by MOS schemes based on zero mean and standardized covariance analysis.

3.5.1.2.3 File Download (Start Time, End Time and File System Location), i.e. download requests in online mode, evaluated by the mobile client. The file download feature shall be modeled as the matrix of number downloads by file location over time, in order to perform malicious behaviors analysis, such as:

1. massive data leakage or similar attacks, which can be identified by MOS schemes based on zero mean covariance analysis;

2. low number of file download from several folders, characterized by sparse operations, which can reveal unreachable navigation performed by automated file download in order to avoid the massive file download characterization. Malicious sparse file download can be identified by MOS schemes based on zero mean and standardized covariance analysis.

3.5.1.2.4 File Upload (Start Time, End Time and File System Location), i.e. upload requests in online mode, evaluated by the mobile client. The file upload feature can reveal attempts of ransomware or similar attacks and be identified by MOS schemes based on covariance analysis. Therefore, it is necessary model the matrix of number uploads by file location over time, in order to perform malicious behaviors analysis, such as:

1. massive file upload, similar to ransomware attack, which can be identified by MOS schemes based on zero mean covariance analysis;

2. low number of file upload to several folders, characterized by sparse operations, which can reveal unreachable navigation performed by automated file upload in order to avoid the massive file upload characterization. Malicious sparse file upload can be identified by MOS schemes based on zero mean and standardized covariance analysis.

3.5.2 Performance analysis

The proposed concept of mobile client security has been implemented in the Storgrid protected cloud environment [92]. Therefore, the approach is correlated with the practical usability requirements: the corporate user continues to use the mobile storage app in offline and does not need to reload the files every time the key is renewed. This methodology can be used in other mobile clients. The common advantage is that the mobile client performs the operations both in the offline and online mode and uses the key expiry to protect the privacy of the corporate data.

The log analysis of the Log Analysis Center (LAC) has been implemented and evaluated for offline anomaly detection in mobile clients, making it possible to apply anomaly detection techniques in a lightweight fashion, considering low processing requirements for deal with the resource constraints of mobile clients. The evaluation considered the required processing time for anomaly detection from log analysis, measuring the data modeling time through the UAL, the eigenvalue decomposition time and the required time for the EDC MOS scheme execution, which is the scheme that requires less processing capacity and provides more anomaly identification accuracy [54, 35].

The experiments were performed in two mobile devices, Galaxy GT-I9300 and Galaxy Tab SM-T800, with variations of log size and window size. The Galaxy GT-I9300 has Quad-core 1.4 GHz Cortex-A9 processor and 1 GB RAM, while the Galaxy Tab SM-T800 has its processing capacity composed by Quad-core 1.9 GHz Cortex-A15 and quad-core 1.3 GHz Cortex-A7, and 3 GB RAM.

Table 3.1 presents the data modeling time and the processing time of eigenvalues decomposition calculations to be applied to anomaly detection from user operation logs of Storgrid mobile client.

The information presented by column are the device model, the log size in megabytes, the window size in minutes, the data modeling time in milliseconds, the average of eigenvalue decomposition time in milliseconds, the standard deviation of eigenvalue decomposition time in milliseconds, the minimum of eigenvalue decomposition time in milliseconds and the maximum of eigenvalue decomposition time in milliseconds.

The results show that the lower window size leads to the larger eigenvalue decomposition time, but the largest eigenvalue decomposition time, which was the maximum of 421 milliseconds with average of 347.42 milliseconds. This result highlights an acceptable speed even for the worst evaluated scenario, which is the case using a Galaxy GT-I9300 for processing 6MB with window size of 10 minutes.

Table 3.2 presents the processing time of EDC MOS calculations applied to anomaly

Table 3.1 Data Modeling and Eigenvalue Decomposition Time

Device	Log Size (MB)	Window (min)	Modeling (ms)	Avg. Eig. (ms)	Std. Eig. (ms)	Eig. Min. (ms)	Eig. Max. (ms)
Galaxy GT-I9300	6	60	107	209.52	18.58	183	276
Galaxy GT-I9300	6	40	115	227.26	18.13	191	289
Galaxy GT-I9300	6	20	89	268.14	21.94	229	315
Galaxy GT-I9300	6	10	90	347.42	24.11	304	421
Galaxy GT-I9300	4.1	60	20	60.90	15.19	37	106
Galaxy GT-I9300	4.1	40	20	68.72	15.71	43	114
Galaxy GT-I9300	4.1	20	34	89.04	16.78	54	133
Galaxy GT-I9300	4.1	10	21	117.24	14.36	96	171
Galaxy GT-I9300	1.4	60	10	159.82	15.82	125	197
Galaxy GT-I9300	1.4	40	10	168.06	15.90	139	220
Galaxy GT-I9300	1.4	20	11	204.4	20.46	176	269
Galaxy GT-I9300	1.4	10	13	259.00	21.34	220	315
Galaxy Tab SM-T800	6	60	7	59.30	6.55	54	74
Galaxy Tab SM-T800	6	40	8	62.56	7.05	56	80
Galaxy Tab SM-T800	6	20	10	73.28	8.59	65	95
Galaxy Tab SM-T800	6	10	8	93.48	9.13	83	130
Galaxy Tab SM-T800	4.1	60	11	18.64	4.51	16	38
Galaxy Tab SM-T800	4.1	40	11	19.64	5.12	17	38
Galaxy Tab SM-T800	4.1	20	12	25.12	5.55	21	46
Galaxy Tab SM-T800	4.1	10	12	32.32	7.29	27	55
Galaxy Tab SM-T800	1.4	60	4	49.08	6.01	42	62
Galaxy Tab SM-T800	1.4	40	5	51.42	7.36	44	74
Galaxy Tab SM-T800	1.4	20	5	51.12	7.80	54	91
Galaxy Tab SM-T800	1.4	10	7	75.24	7.71	65	90

detection from user operation logs of Storgrid mobile client. Table 3.2 respectively presents the device model, the log size in megabytes, the window size in minutes, the average of EDC calculation time in milliseconds, the standard deviation of EDC calculation time in milliseconds, the minimum of EDC calculation time in milliseconds and the maximum of EDC calculation time in milliseconds.

It is possible to observe that the processing time increases with the window size decreasing, similar to the results for eigenvalue decomposition time. The longest processing time measured is lower than 200 milliseconds, even considering window size of 10 minutes or processing 6 MB of user operation log. This result represents an acceptable processing time for anomaly detection in mobile clients.

3.6 Conclusion and future work

An important security issue faced by corporations that use cloud-based systems is how to provide security mechanisms to support offline corporate mobile clients. Once a mobile client releases the connection with the corporate cloud, no security measure implemented in the cloud infrastructure assures the protection of sensitive data stored in the mobile device. Aware of this problem and its importance, this work presented a proposal to address the offline mobile security problem combining cryptographic methods and an

Table 3.2 EDC MOS scheme processing time for anomaly detection

Device	Log Size (MB)	Window (min)	Avg. EDC. (ms)	Std. EDC. (ms)	Min. EDC. (ms)	Max. EDC. (ms)
Galaxy GT-I9300	6	60	5.27	4.04	3	20
Galaxy GT-I9300	6	40	10.78	6.37	6	34
Galaxy GT-I9300	6	20	32.62	12.44	21	88
Galaxy GT-I9300	6	10	115.08	17.45	88	158
Galaxy GT-I9300	4.1	60	5.68	4.18	3	23
Galaxy GT-I9300	4.1	40	10.76	5.31	7	27
Galaxy GT-I9300	4.1	20	37.58	10.30	23	61
Galaxy GT-I9300	4.1	10	125.98	18.56	101	191
Galaxy GT-I9300	1.4	60	4.92	3.49	3	17
Galaxy GT-I9300	1.4	40	9.00	4.23	6	25
Galaxy GT-I9300	1.4	20	30.14	9.21	19	62
Galaxy GT-I9300	1.4	10	100.62	15.83	69	163
Galaxy Tab SM-T800	6	60	1.84	0.65	1	3
Galaxy Tab SM-T800	6	40	3.26	1.24	2	7
Galaxy Tab SM-T800	6	20	10.90	2.40	9	21
Galaxy Tab SM-T800	6	10	41.86	7.33	34	60
Galaxy Tab SM-T800	4.1	60	1.85	0.60	1	3
Galaxy Tab SM-T800	4.1	40	3.62	1.10	2	8
Galaxy Tab SM-T800	4.1	20	12.04	2.79	9	22
Galaxy Tab SM-T800	4.1	10	40.16	6.48	35	60
Galaxy Tab SM-T800	1.4	60	1.98	0.89	1	6
Galaxy Tab SM-T800	1.4	40	3.30	1.16	2	7
Galaxy Tab SM-T800	1.4	20	10.48	2.90	8	21
Galaxy Tab SM-T800	1.4	10	34.52	4.08	30	45

Eigensimilarity-based behavioral anomaly detection.

As proof of concept, a fully working mobile application was developed to test the proposed security solution and acquired results provide evidence that besides achieving the desired security features, the solution also has positive results in terms of performance. This fact is due to the usage of lightweight operations and the optimized combination of the selected security methods. The proposed approach is a practical application to be used in the corporate mobile environment. It is implemented as a fully working mobile client and can be used for any type of enterprise. Also, part of concept is seed for new security solutions for big data apps [93, 94].

4

Robust Framework for Detection of Network Attacks in Imbalanced Traffic

According to [95], 37.9% of all Internet traffic of 2018 was not from human activities, but from bots, that can be classified as good bots, that perform legitimate operations, and bad bots, which perform malicious activities, such as DDoS attacks, probe attack or frauds. In 2018, bad bots accounted for 20.4% of all website traffic.

According to [9] the proportion of cost on discovery activities for cybersecurity has increased steadily since 2015. Forensic cyber and user behavior analytics also present an opportunity for cost savings — US$1.72 million — with discovery and investigation activities. However, only 32% of organizations have deployed these technologies enterprise-wide. Accenture consulting argues that 89% of survey respondents believe breakthrough technologies, like machine learning and user behavior analytics, are essential to securing the future of their organizations [21].

To face the adversarial model, network attacks and counter measures of attackers to avoid detection, it is possible to adopt unsupervised or semi-supervised approaches for network anomaly detection, by means of behavioral analysis, where it is not necessary known anomalies for training models [22].

Anomalies in the context of network traffic can be hard to identify and separate from legitimate data due to the rare occurrences of anomalies in comparison to legitimate events. Therefore, anomaly detection algorithms have to be highly discriminating, robust to corruption and able to deal with the imbalanced data problem [27]. Note that data corruption refers to outliers that can be part of the data but not be considered malicious. The imbalanced data problem corresponds to data sets exhibiting significant imbalances of classes or rare events of some classes [28], which can be legitimate or malicious classes in network anomaly detection problems.

Traditionally adopted algorithms for anomaly detection assume a Gaussian or symmetric distributed data [11], however this assumption may not be observed in some real world problems [29], such as the case of network traffic analysis, where network traffic features are usually more characterized by skewed and heavy-tailed distributions [30].

Findings of Benson *et al.* [29] indicate that certain positive skewed and heavy-tailed distributions can model data center switch traffic, and highlights a difference between the data center environment and the wide area network, where the long-tailed Pareto distribution typically shows the best fit [29]. Leon-Garcia [30] also argues that Pareto distribution has been found to capture the behavior of many quantities of interest in the study of Internet behavior. Moreover, Benson *et al.* [29] observes that the Lognormal distribution is the best fit to model arrival processes in a data center.

The findings presented by Benson *et al.* [29] and Leon-Garcia [30] show that the skewness and heavy-tailed distributions may be important for network traffic analysis, and can motivate researches to evaluate the impact of skewed data into algorithms that rely on Gaussian distributed data for network anomaly detection. Additionally, the fitting of network traffic to skewed and heavy-tailed distributions can indicate opportunities to exploit properties and characteristics of the skewness and heavy-tailed distributions to obtain improved classifiers for network anomaly detection.

Network anomaly detection problems are usually characterized by imbalanced data [96]. However, learning algorithms for imbalanced data has been a challenging research topic, considering that the imbalanced data can compromise the performance of most standard learning algorithms, creating bias or unfair weight to learn from the majority class and reducing detection capacity of anomalies that are characterized by the minority class [27]. Therefore, learning methods for imbalanced and skewed data have attracted attention of researchers [31].

Robust subspace learning has been attracting a growing attention of researchers aiming the development of network attack detection systems [97, 98] that rely on behavioral analysis. An outlyingness-approach based on a robust estimator of skewness, combined with robust estimators of location and scale, can be able to flag the outlying measurements [31]. According to Hubert *et al.* [31], when the same methodology would be used with non-robust estimators of location, scale and skewness, the outlyingness-values would be affected by the outliers such that the outlying group could be masked.

Considering that the skewness of anomalous and legitimate traffic can highlight features for improving anomaly detection and network attack detection in imbalanced data, and considering that the distance between robust estimates can be used for network attack detection, we propose the Moment-based Robust Principal Component Analysis (m-RPCA), which is an approach based on distances between moments computed from a robust subspace learned by Robust Principal Component Analysis (RPCA) from contaminated observations, in order to detect anomalies from skewed data and network traffic.

The proposed approach relies on a robust subspace computed from supposed legitimate observations, for estimating the moments to be used for distance analysis. The anomaly detection from contaminated observations evaluate the Mahalanobis distance between the robust moments and new contaminated observations, in a semi-supervised fashion, without the computational cost of new robust subspace learning for anomaly detection from new observations. The m-RPCA can also be computed as an unsupervised algorithm, with subspace learning from the same contaminated data that is the target of the anomaly detection analysis.

We evaluate the accuracy of the m-RPCA for anomaly detection on simulated data sets, with skewed and heavy-tailed distributions, and for network attack detection on CTU-13 data set [99], which is a large data set of legitimate, background and botnet traffic that has been adopted to deal with the lack of up-to-date real-world data sets for anomaly detection systems [36]. The Experimental evaluation compares our proposal to standard and widely adopted algorithms for anomaly detection, which are based on clustering and statistical approaches, and to ROBPCA [100], which is a anomaly detection method that relies on robust estimates with adjusted outlyingness based on robust skewness.

The main contribution of this work is the proposal of a novel semi-supervised and unsupervised method for anomaly detection in skewed and imbalanced data, with results showing improvements through an experimental evaluation on simulated skewed and heavy-tailed distributions and on real data set with legitimate network traffic and botnet attacks.

This chapter is organized as follows. In Section 4.1, a literature review about network anomaly detection, botnet detection, and imbalanced learning is conducted. We present in Section 4.2 the data model and the evaluated data set. In Section 4.3 it is described the proposed approach for network attack detection. We discuss in Section 4.4 the experimental validation, and in Section 4.5 are presented the results of the simulated scenarios and in Section 4.6 we describe the results for anomaly detection on CTU-13 data set. Finally, in Section 4.7 we draw the conclusions and the suggestions for future work.

4.1 Related Works

Network Anomaly detection has emerging as an important approach for securing communication networks and to deal with the increasing number of network attacks. Bhuyan *et al.* [19] provide an overview of facets of network anomaly detection, present attacks normally encountered by network intrusion detection systems, and categorize existing network anomaly detection methods and systems based on the underlying techniques. Ahmed *et al.* [20] present an analysis of four major categories of anomaly detection techniques, which include classification, statistical, information theory and clustering. Moustafa *et al.* [22] discuss aspects of anomaly-based Network Intrusion Detection Systems (NIDSs), describing details of cyber-attacks and new solutions for anomaly detection, and provides a benchmark data sets for training and validating approaches for network anomaly detection.

A botnet is a network of bots, that are compromised machines under the influence of malware (bot). The botnet is commandeered by a botmaster and used as resource for attacks, such as distributed denial-of-service (DDoS) attacks, and fraudulent activities such as spam, phishing, identity theft, and information ex-filtration. We refer to [20] and [22] for an overview of network attacks. The botmaster coordinate a botnet through a command and control (C&C) channel where bots receive commands and synchronize attacks and fraudulent activities. Centralized C&C structures using the Internet Relay Chat (IRC) protocol have been utilized by botmasters for a long time, but other protocols, such as HTTP, and architectures, such as Peer-To-Peer, have also been adopted [12].

According to [95], 37.9% of all Internet traffic of 2018 was not from human activities, but from bots, that can be classified as good bots, that perform legitimate operations, and bad bots, which are responsible by malicious activities, such as DDoS, probe attack or frauds.

Acarali *et al.* [101] surveys network-based detection approaches for HTTP-based botnets, and discuss the traffic-based features used to detect bot traffic and presents an abstraction of the main types of features related to protocols and OSI layers. Wang *et al.* [16] present an analysis based on 50,704 different Internet DDoS attacks originated of 674 botnets from 23 different botnet families with a total of 9,026 victim belonging to 1,074 organizations in 186 countries. Their analysis reveals that geolocation of the attacking sources follows patterns and enables source prediction, and highlights that multiple attacks to the same target also exhibit strong patterns of inter-attack time interval, also presents that there is a trend for different botnets to launch DDoS attacks targeting the same victim, simultaneously or in turn.

The BotHunter was proposed by Gu *et al.* [102] to detect the infection and coordination of botnets by matching sequence model, through a correlation approach for detecting stages of the infection process. Gu *et al.* [12] presented the BotMiner, which aims to detect groups of compromised machines that are part of a botnet. BotMiner monitors communications that may suggest C&C or malicious activities, and finds a coordinated group pattern by means of clusters of similar communication activities, clusters of similar malicious activities, and performs cross cluster correlation to identify the hosts that share both similar communication patterns and similar malicious activity patterns.

Khattak *et al.* [13] proposed BotFlex, which is a network-based tool for botnet detection, composed by a Complex Event Processing (CEP) engine and on a correlation framework that continuously receives events and correlates them according to rules. BotFlex's results are compared to BotHunter [102], but the evaluation relies in an own and not public data set.

Several approaches for network attack detection uses the KDD 99 [20, 36, 19] data sets for accuracy and performance evaluation, due to their availability and labeled attacks. Even though the KDD 99 data set are criticized by the generation procedure and the risk of over-estimations of anomaly detection due to data redundancy, it still represents one of the few publicly available labeled data sets currently in use today by researchers [36, 19]. NSL-KDD [46] data set is the refined version of the KDD 99 data set that removed the redundant data records, in order to avoid biased classifications. However, NSL-KDD data set maintain the limitations of the KDD 99 regarding volume and lacks on reproduction of recent network traffic and attacks.

The objective of Garcia *et al.* [99] is to compare three botnet detection methods by means of a simple and reproducible methodology, by a good data set and by a new error metric. This paper evaluates some data sets for network anomaly detection, and surveys some approaches for botnet detection, and proposes two methods (BClus and CAMNEP) for botnet detection, comparing results to BotHunter [102].

Considering the lack of available labeled data sets, Garcia *et al.* [99] proposes the CTU-13 data set, which is composed by attack, legitimate and background labeled data, in an imbalanced distribution like in a real networks. The authors recommend scenarios for training and testing in order to avoid the use of traffic from a botnet family for training and testing, aiming to ensure that the evaluated methods can generalize and detect new behaviors. Taking into account the adoption of unsupervised or semi-supervised approaches for anomaly detection, adopting the training and testing approach proposed by Garcia *et al.*, some botnet malwares wouldn't be tested, since in the author's proposal some botnets are present only for training.

Wang and Paschalidis [15] proposed a botnet detection approach based on anomaly and community detection, aiming for detecting botnets and identifying bots before the botnet becomes active. The first stage detects anomalies by leveraging large deviations of an empirical distribution. The second stage detects the bots using ideas from social network community detection in a graph that captures correlations of interactions among nodes over time. This work is compared to the BotHunter [102] for the CTU-13 botnet data set [99].

Traditional PCA-based anomaly detection models are not suitable for anomaly interpretation, as they judge whether a data instance is an anomaly or not based on the length of its projection on the abnormal subspace spanned by the less significant principal components, and there is no direct mapping between PCA's dimensionality-reduced subspace and the original feature space [103]. However, to overcome the above mentioned limitations, some approaches based on PCA have been proposed for network anomaly detection.

Callegari *et al.* [24] proposed a PCA-based method for identifying the network traffic flows responsible for an anomaly detected at the aggregate level, by means of a separation of legitimate and anomaly observations according to principal components (legitimate) and remaining (anomalies). Lee *et al.* [42] presented OverSampling PCA (osPCA), which allows one to determine the anomaly of the target instance according to the variation of the resulting dominant eigenvector obtained by similarity analysis and over sampling. Vieira *et al.* [1] proposed a framework that applies Model Order Selection (MOS) for detection of time frames under attack and uses similarity analysis to extract details and detect the time and ports under attack.

The problem of PCA or subspace learning for outlier corrupted data is called Robust Principal Component Analysis (RPCA) or robust subspace learning [104, 105]. RPCA aims to be resilient to outliers by means of a robust subspace learning [105] for outlier corrupted data, decomposing a given data matrix $\boldsymbol{X}$ into the sum of a low rank matrix $\boldsymbol{L}$, whose column subspace gives the principal components, and a sparse matrix $\boldsymbol{S}$, which refers to outliers' matrix. We refer to [106] and [105] for more details regarding robust subspace learning.

RPCA has been mainly applied to computer vision, in problems of robust subspace tracking and robust subspace recovery. However, RPCA has also been adopted for general outlier detection [100, 31, 107, 98, 108] and for anomaly detection on network traffic

[97]. ROBPCA [100] intends to identify outliers using PCA from robust estimates of mean and covariance matrix, to reduce the data dimensions and plotting the orthogonal distances versus the robust score distances, to flag an outlier map. However, ROBPCA flags many points as outlying when the original data is skewed. Therefore, Hubert *et al.* [31] proposed ROBPCA-AO, which improves ROBPCA for problems with skewed data, by means of an adjusted outlyingness based on robust skewness. Hubert *et al.* [31] evaluated ROBPCA-AO for real and simulated data sets, but it is not clear if this method was evaluated for very skewed and imbalanced data sets, such as in the network attack detection problem.

Robust subspace learning has received a growing attention of researchers aiming the development of network anomaly detection systems [31, 97, 98], considering outlier-robust methods and sparse-corruption methods [106]. Pascoal *et al.* [97] proposed an approach based on a robust mutual information estimator for feature selection and based on RPCA for outlier detection in internet traffic. The anomaly detection proposed by Pascoal *et al.* is an unsupervised approach that estimate the first k robust principal components, calculate the score and the orthogonal distances, calculate the thresholds and classify new observations accordingly.

Zhou and Paffenroth [98] proposed the Robust Deep Autoencoders (RDA), which the central idea is that a RDA inherits the non-linear representation capabilities of autoencoders combined with the anomaly detection capabilities of RPCA. Considering that outliers and noise may reduce the quality of representations discovered by deep autoencoders, the proposed model isolates noise and outliers in the input by means of a RPCA approach, and the autoencoder is trained after this isolation. RDA was evaluated by the authors for the MNIST data set.

Benson *et al.* [29] conducted an empirical study of the network traffic in 10 data centers belonging to three different types of organizations, including university, enterprise, and cloud data centers. Findings of Benson *et al.* indicate that certain positive skewed and heavy-tailed distributions can model data center switch traffic, and highlights a difference between the data center environment and the wide area network, where the long-tailed Pareto distribution typically shows the best fit [29].

Mahalanobis Distance (MD) is a generalized distance which is useful for determining the similarity between an unknown sample and a collection of known samples, by considering the covariance between the variables and their mean values. The MD is a measure of the distance between a vector $\boldsymbol{x}$ and a distribution $\boldsymbol{X}$, introduced by P. C. Mahalanobis in 1936 [109]. MD is a multi-dimensional generalization for measuring how many standard deviations away $\boldsymbol{x}$ is from the mean $\boldsymbol{\mu}$ and covariance $\hat{\boldsymbol{\Sigma}}$ of $\boldsymbol{X}$. MD has been used for distance based anomaly detection with robust estimates in many areas, assuming that the Mahalanobis Distance between robust estimates and contaminated observations can reveal anomalies.

We propose an approach based on the distances between moments computed from learned robust subspace and contaminated observations, for anomaly detection on skewed and imbalanced data sets, and evaluate our approach for network attack detection. The proposed approach relies on learning robust subspace from supposed legitimate traffic for estimating the moments (mean, skewness and kurtosis).

Thus, the anomaly detection for contaminated observations should evaluate the distance between the robust moments and contaminated observations, in a semi-supervised fashion and without the computational cost of new robust subspace learning for new observations, or in an unsupervised approach, where the robust subspace is learned from the contaminated data and the distance analysis is between the robust moments and the contaminated observations.

4.2 Data Model

In this chapter, we adopt the data model introduced in Section 2.2 and define that the Frobenius norm is denoted as $\|\cdot\|_F$, while $\|\cdot\|_*$ denotes the nuclear norm of a matrix and $\|\cdot\|_1$ means the sum of the absolute values of matrix entries. We also define the operator $\langle\cdot\rangle$, that is the standard trace inner product, and the operator $[\cdot]^c$, which denotes the indexes of the c largest values of a vector.

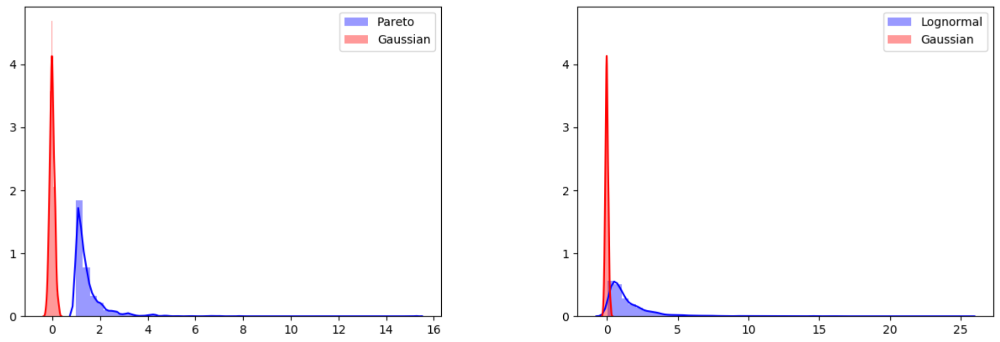
This section also presents a description of the simulated data model in Subsection 4.2.1 and in Subsection 4.2.2 we describe the data model for the CTU-13 data set.

4.2.1 The Simulated data set

In order to analyze the hypotheses $H_2^{(N)}$ and $H_2^{(A)}$, which evaluate if the robust subspace learning can improves the anomaly detection from imbalanced and skewed data, we create two simulated data sets characterized by skewed and heavy tailed distributions, and create a simulated Gaussian distributed data set, in order to also analyze the detection rate for not skewed and not heavy tailed distributions.

We selected Pareto (with scale $a = 3$, mode $m = 1$ and $\mu = 1$) and Lognormal (with mean $\mu = 0$ and standard deviation $\sigma = 1$) distributions, denoted respectively as $\boldsymbol{Y}_p$, while $\boldsymbol{Y}_l$, to simulate legitimate events and to evaluate the anomaly detection on skewed data performed by our proposals in comparison to widely adopted algorithms for outlier detection, considering that these distributions are skewed and heavy-tailed, and have been adopted to characterize network traffic of the Internet and data centers [29, 30]. We also adopt the Gaussian distributed data set $\boldsymbol{Y}_g$ to simulate legitimate events with mean $\mu = 0$ and unitary standard deviation $\sigma = 1$.

The addictive anomalies for Pareto and Lognormal distributions are white Gaussian noise with mean $\mu = 0$ and unitary standard deviation $\sigma = 1$. The Pareto distribution contaminated by Gaussian noise is denoted as $\boldsymbol{Y}_p^c$, while $\boldsymbol{Y}_l^c$ denotes the Lognormal distribution contaminated by Gaussian noise. The $\boldsymbol{Y}_p^c$ and $\boldsymbol{Y}_l^c$ are depicted by the Figures 4.1(a) and 4.1(b), in order to provide a visual example one variable of $\boldsymbol{Y}_p^c$ and $\boldsymbol{Y}_l^c$.



(a): Pareto and Gaussian Anomalies

(b): Lognormal and Gaussian Anomalies

Figure 4.1 Examples of skewed and heavy tailed distributions

An uniform noise between -6 and 6 is used to add anomalies into the Gaussian distributed data set, and $\boldsymbol{Y}_g^c$ denotes the contaminated Gaussian data contaminated by an uniform distribution. The figure 4.2 shows an example of the Gaussian distribution of legitimate observations and the uniform distribution to simulate the addition of anomalies.

Each contaminated data set is composed by a number of legitimate observations and contaminated samples. We evaluate contamination rates c between 1% and 50%, to simulate the imbalanced data of anomaly detection problems during our experiments. This data sets simulate a total of 2400 events for each scenario with contaminated Pareto, Lognormal and Gaussian. Therefore, the number of legitimate observations is defined according to the contamination rate selected for each evaluation.

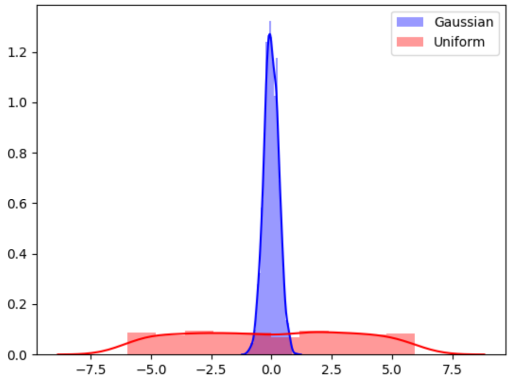


Figure 4.2 Example of Gaussian and Uniform Anomalies

4.2.2 The CTU-13 data set

The CTU-13 [99] is a data set of botnet traffic that was captured in the Czech Technical University, by means of a testbed and malware execution in a real network. The CTU-13 data set contains 13 scenarios with network flows of botnet malwares, that are: neris, rbot, virut, menti, sogou, nsys.ay and murlo. The botnet traffic is also classified as attack or command and control (C&C), while the legitimate flows can also be classified as legitimate or background.

The types of C&C and attack flows present in CTU-13 data set are:

- **Attacks:** Click Fraud (CF), Port Scan (PS), Fast Flux (FF), SPAM and DDoS;
- **C&C:** IRC, P2P and HTTP.

We refer to Garcia [110] and Garcia *et al.* [99] for a detailed description of the performed attacks and C&C flows, as well as for more information about the topology of the adopted testbed, rules for classifying legitimate flows, and an analysis of behaviors or patterns of the malware's traffic.

For all the scenarios, the authors of the CTU-13 data set convert the captured pcap files to NetFlows and release the processed flows. The data set provides ground-truth labels for flows as follows: flows from or to the infected machines are labeled as "botnet"; flows from or to well-known and controlled machines are labeled as "normal"; all other flows are labeled as "background."

Table 4.1 presents an overview grouped by scenario, according to the column ID, and shows the malwares used for botnet attacks, the types of attacks or C&C, the total number of flows, the number of malicious flows which includes flows of C&C and attacks, and finally shows the number of legitimate flows.

Table 4.1 CTU-13 data set description

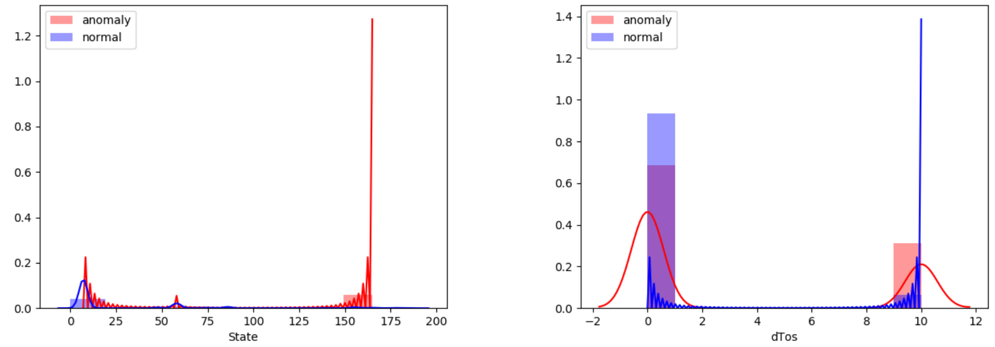
ID	Bot	Type	Total	Malicious	C&C	Attack	Normal
10	neris	IRC, Spam, CF	2,824,636	40,961 (1.45%)	341 (0.01%)	40,620 (1.44%)	30,387 (1.07%)
11	neris	IRC, Spam, CF	1,808,122	20,941 (1.16%)	673 (0.04%)	20,268 (1.12%)	9,120 (0.5%)
12	rbot	IRC, PS	4,710,638	26,822 (0.57%)	63 (0.00%)	26,759 (0.57%)	116,887 (2.48%)
15	rbot	IRC, DDoS	1,121,076	2,580 (0.23%)	52 (0.00%)	2,528 (0.23%)	25,268 (2.25%)
15-2	virut	Spam, PS, HTTP	129,832	901 (0.69%)	24 (0.02%)	877 (0.68%)	4,679 (3.6%)
16	menti	PS	558,919	4,630 (0.83%)	199 (0.04%)	4,431 (0.79%)	7,494 (1.34%)
16-2	sogou	HTTP	114,077	63 (0.06%)	26 (0.02%)	37 (0.03%)	1,677 (1.47%)
16-3	murlo	PS	2,954,230	6,127 (0.21%)	1,074 (0.04%)	5,053 (0.17%)	72,822 (2.46%)
17	neris	IRC, Spam, CF, PS	2,087,508	184,987 (8.86%)	2,973 (0.14%)	182,014 (8.72%)	43,340 (1.57%)
18	rbot	IRC, DDoS	1,309,791	106,352 (8.12%)	33 (0.00%)	106,319 (8.12%)	15,847 (1.2%)
18-2	rbot	IRC, DDoS	107,251	8,164 (7.61%)	2 (0.00%)	8,162 (7.61%)	2,718 (2.53%)
19	nsys.ay	P2P	325,471	2,168 (0.67%)	25 (0.01%)	2,143 (0.66%)	7,628 (2.35%)
15-3	virut	Spam, PS, HTTP	1,925,149	40,003 (2.08%)	536 (0.03%)	39,467 (2.05%)	31,939 (1.65)

The full data set and scenarios can be denoted as $\boldsymbol{X} = \{\boldsymbol{X}_{10}, \boldsymbol{X}_{11}, \ldots, \boldsymbol{X}_{18-2}, \boldsymbol{X}_{19}\}$, in accordance to IDs presented in Table 4.1. In our experiment each contaminated scenario $\boldsymbol{X}_i$ is split into $\boldsymbol{X}_i^s$ containing 50% of the legitimate data, and into $\boldsymbol{X}_i^c$ that is composed by all anomalous flows and the necessary number of legitimate flows to have a testing data with the desired contamination rate.

The CTU-13 data set originally contains the following features for each flow: Start Time, End Time, Duration, Source IP Address, Source Port. Direction, Destination IP Address, Destination Port, State of TCP flags, Destination Type of Service, Source Type of Service, Total number of Packets, Total number of Bytes.

Our analysis of the available features leads to discard some features, considering that highly correlated features can bias or not improve the model, and that source or destination IP addresses can insert some false bias into learning models, since they can be changed by IP spoofing. Other risk related to adopt IP address for training models is the training model to learn that one IP is legitimate and this IP be infected subsequently, which can result into false negative classifications.

We conducted and Exploratory Data Analysis (EDA) on the CTU-13 and observed that some features are skewed and present high overlapping between legitimate and anomalous flows, as can be seen in Figure 4.3(a) and Figure 4.3(b), that present the distributions of TCP state of scenario 16 and the type of services from destination of scenario 10.

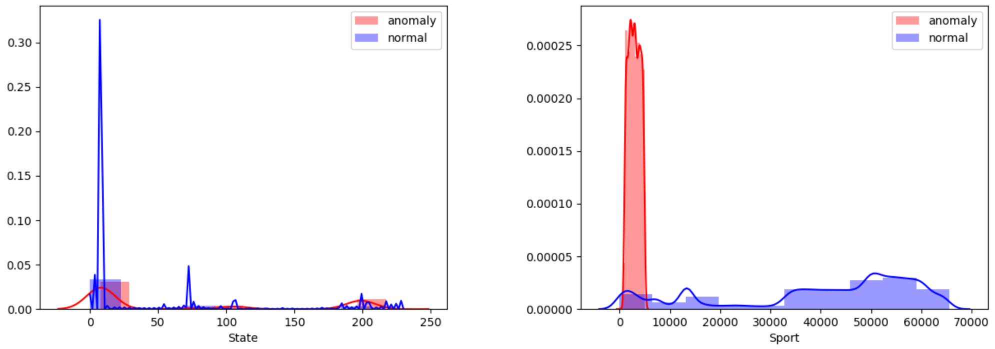


(a): State of scenario 16.

(b): Destination Type of service of scenario 10.

Figure 4.3 Example of skewness and overlapping

However, it is not possible to observe a pattern on distributions of all the features and scenarios of CTU-13, as depicted by Figures 4.4(a) and 4.4(b), that show the distributions of TCP states of the scenario 10 and source ports of scenario 16, and highlight the distributions of legitimate and anomalous flows.



(a): State of scenario 10.

(b): Source Port of scenario 16.

Figure 4.4 Skewness and Overlapping

Due to the number of available features and the class overlapping between legitimate and anomalous flows, we performed an correlation analysis and an empirical cross validation in order to identify the best set of features for network attack detection. Therefore, we adopt the following features: state, destination type of service, destination port, source port, total number of packets, total number of bytes and number of bytes from the source.

4.3 Moment Distances from Robust Subspace for Network Attack Detection

This section describes the proposed approach for network attack detection by means of a distance analysis between moments computed from a robust subspace and contaminated observations of network traffic.

Robust subspace learning can be defined as the decomposition of a given data matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, with rows representing observations and columns representing features. Note that in Section 2.2.2 we introduced $\boldsymbol{X}$ for a packet level analysis, modeled as a matrix of communication port by time. However, in this chapter we adopt a flow based analysis, where $\boldsymbol{X}$ is modeled as a matrix of flows by features, and $\boldsymbol{X}$ is decomposed into the sum of a low rank matrix $\boldsymbol{L} \in \mathbb{R}^{M \times N}$, whose column subspace gives the robust principal components without outliers and noise, and a sparse matrix $\boldsymbol{S} \in \mathbb{R}^{M \times N}$, with element-wise outliers or noise.

Even though robust subspace learning has been adopted for network anomaly detection by means of highlights from the matrix $\boldsymbol{S}$, it was shown that $\boldsymbol{S}$ can indicate noise and outliers that can not be classified as malicious [105, 106], resulting into false positive classifications and requiring complementary approaches in order to obtain precise network anomaly detection [98].

Therefore, for network anomaly detection, we propose to learn a robust subspace from the legitimate traffic $\boldsymbol{X}^s$ for computing $\boldsymbol{L}^s$ and $\boldsymbol{S}^s$, followed by computing the robust moments, i.e. the mean $\boldsymbol{\mu}$, skewness $\boldsymbol{\varepsilon}$ and kurtosis $\boldsymbol{\kappa}$, in order to evaluate the distance $\boldsymbol{d}$ between contaminated observations $\boldsymbol{X}^c$ and robust moments. The largest distances are classified as anomalous and indicate network attacks, denoted as $\boldsymbol{N}$ by the data model $\boldsymbol{X} = \boldsymbol{U} + \boldsymbol{N}$, introduced by (2.1), where $\boldsymbol{U}$ denotes the legitimate network traffic.

RPCA is a well-known method to recover a low-rank matrix $\boldsymbol{L}$ and sparse matrix $\boldsymbol{S}$ from corrupted measurements modeled as $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$. This decomposition in low-rank and sparse matrices can be achieved by techniques such as Principal Component Pursuit method (PCP), and by optimization methods, such as the Augmented Lagrange Multiplier Method (ALM), Alternating Direction Method (ADM), Fast Alternating Minimization (FAM) or Iteratively Reweighted Least Squares (IRLS) [104, 105, 106].

According to Wright et al. [111], under rather broad conditions, as long as the error matrix $\boldsymbol{S}$ is sufficiently sparse, it is possible to recover a low-rank matrix by solving the following convex optimization problem:

$$(\hat{\boldsymbol{L}},\hat{\boldsymbol{S}}) \leftarrow \min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 \tag{4.1}$$

$$\text{subject to: } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$$

$$\|\boldsymbol{L}\|_* = \sum_i \sigma_i(\boldsymbol{L}) \tag{4.2}$$

$$\|\boldsymbol{S}\|_1 = \sum_{ij} |\boldsymbol{S}_{ij}| \tag{4.3}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix, λ is a positive weighting parameter, which determines the sparsity of $\boldsymbol{S}$, and $\|\cdot\|_1$ means the sum of the absolute values of matrix entries, and σ denotes the singular values of a matrix.

Before ALM, some methods were proposed to solve that convex optimization problem, such as Iterative Thresholding (IT) and Accelerated Proximal Gradient (APG). However, according to Zhouchen *et al.* [112], both approaches have scalability problems and require a large number of iterations to converge. The Augmented Lagrange Multiplier (ALM) is proven to have a *Q-linear* convergence speed and experimental results show that ALM is five times faster than APG, which in theory is sub-linear [112]. Furthermore, ALM reaches more accurate results with less iterations.

The RPCA with ALM can be formulated as:

$$l(\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}) = \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1 + \langle \boldsymbol{Y}, \boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S} \rangle + \frac{\mu}{2} \|\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}\|_F^2, \tag{4.4}$$

where $\boldsymbol{Y}$ is the multiplier of the linear constraint and μ is the penalty parameter for the violation of the linear constraint [113]. Thus, an iterative scheme can be presented as:

$$\begin{cases} (\boldsymbol{L}_{k+1}, \boldsymbol{S}_{k+1}) \in \underset{\boldsymbol{L},\boldsymbol{S} \in \mathbb{R}^{m \times n}}{\arg\min} \{ l(\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}_k) \}, \\ \boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{X} - \boldsymbol{L}_k - \boldsymbol{S}_k), \end{cases} \tag{4.5}$$

We adopt RPCA with ADM, which, generally speaking, is a practical improvement of the classical ALM method for solving convex programming problem with linear constraints, by fully taking advantage of its high-level separable structure [113]. ADM minimizes $\boldsymbol{L}$ and $\boldsymbol{S}$ variables serially by solving the following problems to generate the new iterate:

$$\begin{cases} \boldsymbol{L}_{k+1} \in \underset{\boldsymbol{L} \in \mathbb{R}^{m \times n}}{\arg\min}\{l(\boldsymbol{L}, \boldsymbol{S}_k, \boldsymbol{Y}_k)\} \\ \boldsymbol{S}_{k+1} \in \underset{\boldsymbol{S} \in \mathbb{R}^{m \times n}}{\arg\min}\{l(\boldsymbol{L}_{k+1}, \boldsymbol{S}, \boldsymbol{Y}_k)\} \\ \boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{X} - \boldsymbol{L}_k - \boldsymbol{S}_k) \end{cases} \tag{4.6}$$

Moments are a set of statistical parameters to measure a distribution. The arithmetic mean is the first general moment, the second is the variance, while skewness (asymmetry) is the third moment and kurtosis (tailedness) is the fourth moment [114].

Let the mean $\boldsymbol{\mu} \in \mathbb{R}^{1 \times N}$ be

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_i, \tag{4.7}$$

where M is the number of samples, and let the sample covariance matrix $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{N \times N}$ be

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T, \tag{4.8}$$

According to Zwillinger and Kokoska [115], the general expression for the p-th moment $\boldsymbol{m}_p \in \mathbb{R}^{1 \times N}$ about the mean $\boldsymbol{\mu}$ is given by

$$\boldsymbol{m}_p = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{x}_i - \boldsymbol{\mu})^p. \tag{4.9}$$

Therefore, Zwillinger and Kokoska [115] present that the skewness $\boldsymbol{\varepsilon} \in \mathbb{R}^{1 \times N}$ about the mean $\boldsymbol{\mu}$ is calculated by

$$\boldsymbol{\varepsilon} = \frac{\boldsymbol{m}_3}{\boldsymbol{m}_2^{\frac{3}{2}}}, \tag{4.10}$$

and the kurtosis $\boldsymbol{\kappa} \in \mathbb{R}^{1 \times N}$ is given as

$$\boldsymbol{\kappa} = \frac{\boldsymbol{m}_4}{\boldsymbol{m}_2^2}. \tag{4.11}$$

We propose to compute the mean $\boldsymbol{\mu}$, the skewness $\boldsymbol{\varepsilon}$, the kurtosis $\boldsymbol{\kappa}$ and the covariance matrix $\hat{\boldsymbol{\Sigma}}$ from the robust subspace $\boldsymbol{L}^s$, after to minimize the (4.6) for $\boldsymbol{X}^s$. We also propose to compute the Mahalanobis Distance (MD) for detecting anomalies between contaminated observations and the moments calculated from a robust subspace computed by RPCA.

The classical Mahalanobis Distance is defined as

$$\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\mu},\hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})^T}, \tag{4.12}$$

where $\boldsymbol{x}$ is a vector of a new observations, $\boldsymbol{\mu}$ is the mean vector of known observations, also referred as location, and $\hat{\boldsymbol{\Sigma}}$ is the covariance matrix of known observations, also referred as scatter. The classical MD usually relies on robust mean and robust covariance matrix for outlier detection, which are commonly computed by MCD [116, 117]. Here we propose to compute the Robust-Mean Distance $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\mu},\hat{\boldsymbol{\Sigma}})$ according to (4.12), but adopting the mean $\boldsymbol{\mu}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}$ calculated from a robust subspace $\boldsymbol{L}^s$ learned by RPCA.

We also propose to extend (4.12) to implement the Robust-Skewness Distance $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\varepsilon},\hat{\boldsymbol{\Sigma}})$, as follows:

$$\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\varepsilon},\hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\varepsilon})\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\boldsymbol{\varepsilon})^T}. \tag{4.13}$$

Finally, we propose to extend (4.12) to implement the Robust-Kurtosis Distance $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\kappa},\hat{\boldsymbol{\Sigma}})$, as follows:

$$\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\kappa},\hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\kappa})\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\boldsymbol{\kappa})^T}. \tag{4.14}$$

The distances $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\mu},\hat{\boldsymbol{\Sigma}})$, $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\varepsilon},\hat{\boldsymbol{\Sigma}})$ and $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\kappa},\hat{\boldsymbol{\Sigma}})$ shall be computed and evaluated separately and independently, for network attack detection based on robust mean, robust skewness and robust kurtosis, respectively.

Therefore, the robust subspace $\boldsymbol{L}^s$ learned from legitimate data $\boldsymbol{X}^s$ followed by the Robust-Mean Distance $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\mu},\hat{\boldsymbol{\Sigma}})$ is called as Mean Distance of Robust Principal Component Analysis (md-RPCA), or is called Skewness Distance of Robust Principal Component Analysis (sd-RPCA) when followed by Robust-Skewness Distance, given by $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\varepsilon},\hat{\boldsymbol{\Sigma}})$, or Kurtosis Distance of Robust Principal Component Analysis (kd-RPCA) when followed by Robust-Kurtosis Distance, given by $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\kappa},\hat{\boldsymbol{\Sigma}})$.

The contamination rate, denoted by c, is a parameter traditionally adopted by well established outlier detection algorithms [118], and refers to the percentage rate of observations that are known as anomalous. The contamination rate can be well-known for some areas, or can be computed by cross-validation [119] or can be assumed according to previous observations. In our proposal, the contamination defines the number of the largest distances $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\mu},\hat{\boldsymbol{\Sigma}})$, $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\varepsilon},\hat{\boldsymbol{\Sigma}})$ or $\boldsymbol{d}(\boldsymbol{x},\boldsymbol{\kappa},\hat{\boldsymbol{\Sigma}})$ that shall be classified as anomalous.

The observations classified as legitimate and anomalous, according to the contamination c, are denoted by the vector $\hat{\boldsymbol{t}}$, with values of 1 for observations classified as anomalous and 0 to denote legitimate observations.

The Algorithm 2 describes all possible steps of m-RPCA and the approaches md-RPCA, sd-RPCA and kd-RPCA, for the semi-supervised learning approach. The unsupervised approach adopts the same steps, but adopting the contaminated data for robust subspace learning, and requiring new robust subspace learning for testing anomaly detection on new set of observations.

Algorithm 2: Moment Distances from Robust Subspace

```
    Result: $\hat{\boldsymbol{t}}_\mu, \hat{\boldsymbol{t}}_\varepsilon, \hat{\boldsymbol{t}}_\kappa$
 1  Given $\boldsymbol{X}$ split into $\boldsymbol{X}^s$ and $\boldsymbol{X}^c$;
 2  while not $\min_{L,S} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{S}\|_1$ from $\boldsymbol{X}^s$ do
 3      $\boldsymbol{L}_{k+1} \in \underset{\boldsymbol{L} \in \mathbb{R}^{m \times n}}{\arg\min} \{l(\boldsymbol{L}, \boldsymbol{S}_k, \boldsymbol{Y}_k)\}$;
 4      $\boldsymbol{S}_{k+1} \in \underset{\boldsymbol{S} \in \mathbb{R}^{m \times n}}{\arg\min} \{l(\boldsymbol{L}_{k+1}, \boldsymbol{S}, \boldsymbol{Y}_k)\}$;
 5      $\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{X} - \boldsymbol{L}_k - \boldsymbol{S}_k)$;
 6  end
 7  $\boldsymbol{\mu} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{x}_i$, from $\boldsymbol{L}$;
 8  $\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$, from $\boldsymbol{L}$;
 9  $\boldsymbol{\varepsilon} = \frac{\boldsymbol{m}_3}{\boldsymbol{m}_2^{\frac{3}{2}}}$, from $\boldsymbol{L}$;
10  $\boldsymbol{\kappa} = \frac{\boldsymbol{m}_4}{\boldsymbol{m}_2^2}$, from $\boldsymbol{L}$;
11  $\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu}) \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})^T}$, from $\boldsymbol{X}^c$;
12  $\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\varepsilon}, \hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\varepsilon}) \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \boldsymbol{\varepsilon})^T}$, from $\boldsymbol{X}^c$;
13  $\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\kappa}, \hat{\boldsymbol{\Sigma}}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\kappa}) \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \boldsymbol{\kappa})^T}$, from $\boldsymbol{X}^c$;
14  $\hat{\boldsymbol{t}}_\mu = [\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}})]^c$;
15  $\hat{\boldsymbol{t}}_\varepsilon = [\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\varepsilon}, \hat{\boldsymbol{\Sigma}})]^c$;
16  $\hat{\boldsymbol{t}}_\kappa = [\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{\kappa}, \hat{\boldsymbol{\Sigma}})]^c$;
```

The steps between 1 and 10 of the Algorithm 2 are the training from legitimate data $\boldsymbol{X}^s$, which are common steps shared by md-RPCA, sd-RPCA and kd-RPCA. The steps between 11 and 16 aim the anomaly detection from new contaminated observations, by means of Mahalanobis Distance of robust moments. Note that the steps 11 and 14 refer to the steps of md-RPCA for anomaly detection, while the steps 12 and 15 refer to the sd-RPCA, and the steps 13 and 16 refer to the steps of kd-RPCA. It is important to highlight that the anomaly detection from new observations does not require new robust subspace learning when adopting the semi-supervised approach, which only requires to compute a moment-based distance to classify the c largest distances as anomalous or legitimate.

The m-RPCA can also be adopted as unsupervised algorithm if the lines 1 and 2 of the Algorithm 2 are changed to substitute $\boldsymbol{X}^s$ by $\boldsymbol{X}^c$. Hence, the robust subspace is learned from contaminated data and used for comparing the distance between moments from robust estimate and contaminated observations. It is important to note that this unsupervised approach requires the computational cost of computing new subspace learning for any new set of observations.

4.4 Experiments

This section presents the performed experiments on simulated and real data set for anomaly detection. First, in Section 4.4.1 we present the adopted metric to evaluate imbalanced data in the context of anomaly detection. In Section 4.4.2 we describe the experiment for anomaly detection on simulated skewed and heavy-tailed data set, and we present in Section 4.2.2 the experiment for network anomaly detection on CTU-13 data set.

4.4.1 The metric

In anomalous detection problems, where anomalies are rare events, if one classify all observations as legitimate and apply an accuracy evaluation, one would have high accuracy but poor true-positive detection. Due to the importance given by the F_1 (also referred as F-score or F-measure) to true-positive detection in scenarios such as the network attack detection, it is the preferable measure for imbalanced data sets [120, 22]. Therefore, F_1 is the metric used for validation of our experiments.

The F_1 is the harmonic mean of precision and recall, where p_p denotes the precision and recall is denoted as r_c. The F_1 is given by

$$F_1 = 2 \cdot \frac{p_\mathrm{r} \cdot r_\mathrm{c}}{p_\mathrm{r} + r_\mathrm{c}}. \tag{4.15}$$

Precision can be seen as a measure of exactness or accuracy, that relies on true positive and false positive measures, denoted by t_p and f_p, respectively. The precision is defined by

$$p_r = \frac{t_\mathrm{p}}{t_\mathrm{p} + f_\mathrm{p}}. \tag{4.16}$$

Recall is a measure of completeness, to calculate proportion of actual positives was identified correctly, by means of the true positive and false negative measures, denoted by t_p and f_n. The recall is defined by

$$r_c = \frac{t_\mathrm{p}}{t_\mathrm{p} + f_\mathrm{n}}. \tag{4.17}$$

This experimental evaluation compares our proposal to widely adopted algorithms for anomaly detection that also rely on contamination rate for anomaly detection, which are: A PCA approach based on the sum of weighted projected distances to the eigenvector hyperplanes [121]; MCD [116, 117]; One-Class Support Vector Machines [122]; Local Outlier Factor (LOF) [123]; k-Nearest Neighbors [124]; and Isolation Forest [125].

We also compare the results of our proposals to ROBPCA for anomaly detection on simulated and CTU-13 data sets, considering that ROBPCA also relies on robust estimates with adjusted outlyingness based on robust skewness [31].

4.4.2 Simulated Experiment

Anomaly detection algorithms usually rely on supervised or unsupervised methods, where the former requires labeled legitimate and anomalous data for training anomaly detection models, while the latter does not require labeled data or training. Semi-supervised algorithms are an alternative for the anomaly detection problem, considering that this method only relies on legitimate data for training and that non-malicious data can be obtained from historical information and from rule-based approaches.

We propose semi-supervised and unsupervised approaches for m-RPCA, where the former relies on legitimate data $\boldsymbol{X}^s$ for training and on contamination rate c for anomaly detection, while the latter relies only on contaminated data $\boldsymbol{X}^c$ for robust subspace learning and relies on contamination rate c to select the largest distances. We assume that c is well-known or can be estimated for real world problems of anomaly detection, in accordance to the assumption adopted by the well established algorithms [118] selected for comparison, that also rely on contamination rate $\boldsymbol{X}^c$.

The availability of labeled data is a challenging concern in real world problems of anomalous detection, where anomalies are rare or even unknown events. Considering that RPCA has already been adopted to isolate outliers from training data [98], we also propose to evaluate the m-RPCA for a contaminated semi-supervised approach based on training from contaminated data, in order to evaluate the impact that contaminated data for robust subspace learning can cause in the anomaly detection results.

Therefore, we propose to evaluate the following approaches for m-RPCA: semi-supervised, contaminated semi-supervised, and unsupervised.

For the semi-supervised approach we propose to learn the robust subspace and compute the moments from the legitimate data sets $\boldsymbol{Y}_g$, $\boldsymbol{Y}_p$ and $\boldsymbol{Y}_l$ with Gaussian, Pareto and Lognormal distributions, respectively, and test the anomaly detection for contaminated data sets $\boldsymbol{Y}_g^c$, $\boldsymbol{Y}_p^c$ and $\boldsymbol{Y}_l^c$.

For the Contaminated Semi-supervised approach we evaluate the robustness of the m-RPCA approaches for learning from contaminated training data, in order to analyze if m-RPCA can be an alternative for the lack of known legitimate data. Therefore we propose to train the model from a contaminated legitimate data $\boldsymbol{Y}_g^{c'}$, $\boldsymbol{Y}_p^{c'}$ and $\boldsymbol{Y}_l^{c'}$, with the same contamination rate of the testing data, but without data repetition between training and testing, taking into account that we shall consider a different contaminated data but adopt the same contamination rate for training and testing.

We finally evaluate the unsupervised approach, that relies on the test data sets $\boldsymbol{Y}_g^c$, $\boldsymbol{Y}_p^c$ and $\boldsymbol{Y}_l^c$ for robust subspace learning, and then classify the results according to the distance between the contaminated data $\boldsymbol{Y}_g^c$, $\boldsymbol{Y}_p^c$ and $\boldsymbol{Y}_l^c$, and the moments computed from the learned robust subspace.

4.4.3 Experiment for CTU-13

In contrast to Garcia *et al.* [99], we propose to evaluate each scenario of the CTU-13 data set individually, in a semi-supervised approach that does not rely on training data with labeled anomalies, in order to evaluate our proposed approaches to all botnet malwares of the CTU-13 data set.

In our experiment setup each contaminated scenario $\boldsymbol{X}_i$ of CTU-13 is split into $\boldsymbol{X}_i^s$, containing 50% of the legitimate data, and into test data $\boldsymbol{X}_i^c$ containing 33% of legitimate and 67% of anomalous data. We adopt a contamination rate c of 33% for experimenting anomaly detection for the CTU-13 data set, considering that this contamination rate presented good results on the simulated experiment for our proposals and for some selected algorithms.

This experiment consider only the semi-supervised approach due to results of simulated experiments on simulated data set presented in Subsection 4.5, that shows better results for the semi-supervised approach and highlight that the this approach can obtain good results even when trained with contaminated data set.

4.5 Results of Simulated Experiment

We adopt prefixes to denote the evaluated approaches for md-RPCA, sd-RPCA and kd-RPCA, which are ss_ to denote semi-supervised, css_ to denote contaminated semi-supervised and u_ for unsupervised.

We evaluated the F_1 of the selected algorithms and m-RPCA approaches for anomaly detection on Gaussian distributed legitimate data contaminated by uniform distributed anomalies. The Figure 4.5 shows the F_1 over the contamination rate between 1% and 50% for m-RPCA approaches and Isolation Forest (IF) [125], k-Nearest Neighbors (KNN) [124], Local Outlier Factor (LOF) [123], Minimum Covariance Deteminant (MCD) [117], One-Class Support Vector Machines (OCSVM) [122], PCA [121] and ROBPCA-AO [31].

It is possible to observe in Figure 4.5 that LOF, PCA, css_kd-RPCA and u_kd-RPCA presented lower performance than the remain algorithms, that obtain results higher than 0.95 in average. The exception is the ROBPCA-AO, that presented high score for lower contamination but decreased with the contamination increasing.

Note that the css_kd-RPCA and u_kd-RPCA are the contaminated semi-supervised and unsupervised versions of kd-RPCA, that obtain worse results than the semi-supervised approach of kd-RPCA, for anomaly detection on Gaussian distributed data contaminated by uniform anomalies. However, the unsupervised versions of md-RPCA and sd-RPCA presented similar results to widely adopted unsupervised algorithms for outlier detection. The results also show that the semi-supervised approach of md-RPCA, sd-RPCA and kd-RPCA obtain high anomaly detection rate and presented similar results to the unsupervised algorithms IF, KNN, MCD and OCSVM.

The Figure 4.6 and 4.7 show the results for anomaly detection on skewed and heavy tailed distributions. The Figure 4.6 shows the results for Pareto with Gaussian anomalies.

The results for anomaly detection on Pareto with Gaussian anomalies, depicted by Figure 4.6, show that IF, KNN, LOF, OCSVM, css_md-RPCA and u_md-RPCA performed worse than the remain algorithms for the evaluated contamination, with results lower than 0.6 even with higher contamination. Note that the approaches of m-RPCA with lower scores are css_md-RPCA and u_md-RPCA, which are mean-based approaches. However, the ss_md-RPCA is the mean-based approach that presented lower results for lower contamination but achieved more than 0.8 with contamination near of 0.2 or higher,

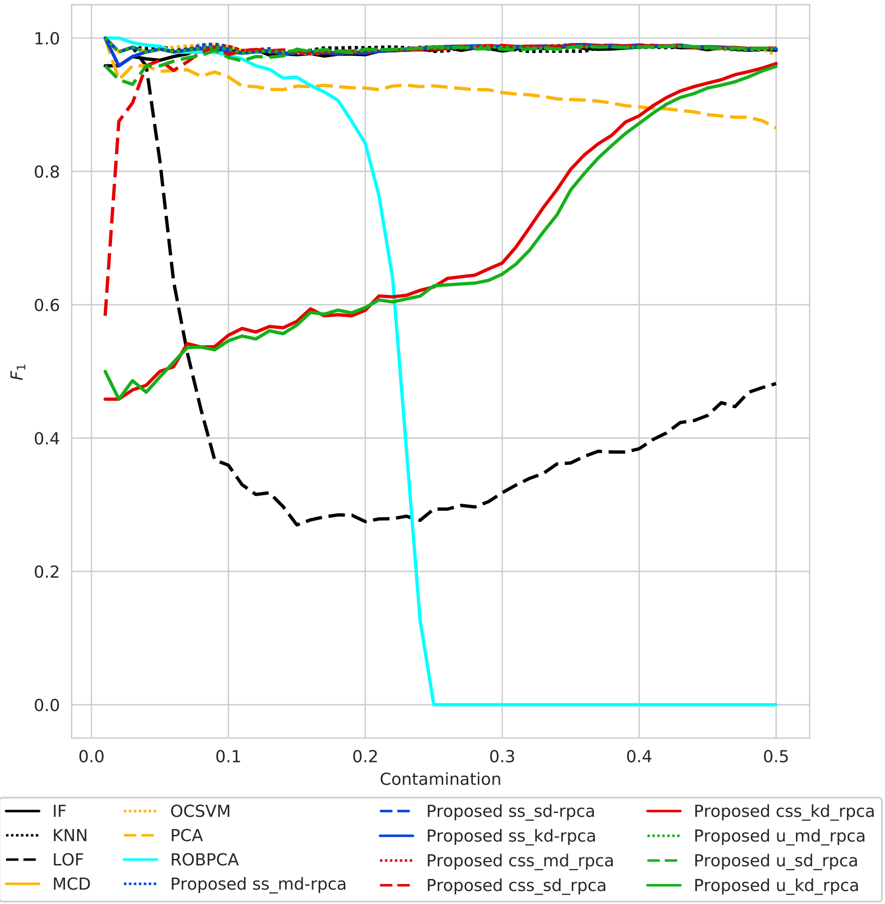


Figure 4.5 Anomaly detection on Gaussian distributed with uniform anomalies

and achieved results better than MCD and PCA.

Figure 4.6 shows that ROBPCA-AO presented high scores for low contamination rates, presenting results better than ss_md-RPCA, MCD and PCA, initially. However, the results of ROBPCA-AO decrease drastically with the contamination increasing.

It is possible to observe in Figure 4.6 that all approaches based on kd-RPCA achieved the best results initially, but the results for the unsupervised variate with contamination near of 0.2 or higher, while the ss_kd-RPCA presents stable results near of 1.0 F_1 for all evaluated contamination rates.

All approaches based on sd-RPCA obtained anomaly detection higher than 0.8, however the unsupervised and contaminated semi-supervised presented high variation of F_1 and lower results in comparison to the semi-supervised sd-RPCA. The contaminated semi-supervised approaches of sd-RPCA and kd-RPCA presented results higher than 0.8 and similar to the semi-supervised and unsupervised approaches of sd-RPCA and

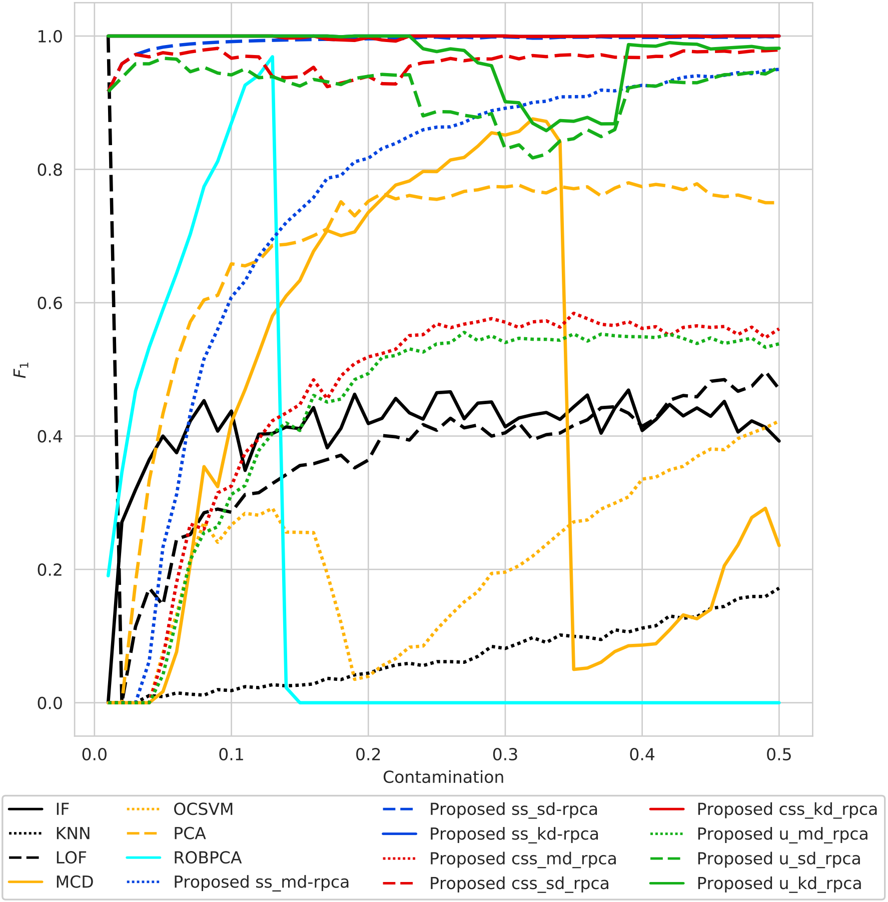


Figure 4.6 Anomaly detection on Pareto distributed with Gaussian anomalies

kd-RPCA. These results highlight the resilience of the robust subspace learning even for contaminated training data.

The unsupervised approaches of sd-RPCA and kd-RPCA presented more than 0.8 for all contamination rate, showing better results than the widely adopted unsupervised algorithms for outlier detection. However, u_kd-RPCA and u_sd-RPCA presented lower results than ss_kd-RPCA and ss_sd-RPCA. Therefore, the semi-supervised approaches of m-RPCA overcome other approaches of m-RPCA and overcome the selected algorithms for anomaly detection on Pareto distributed data with Gaussian anomalies. It is possible to highlight the semi-supervised kd-RPCA, which obtained stable results near of 1.0 F_1 for all evaluated contamination.

The results for anomaly detection on Lognormal with Gaussian anomalies, depicted by Figure 4.7, show that IF, KNN, LOF, MCD, OCSVM, css_md-RPCA, u_md-RPCA and ROBPCA-AO perform worse than the remain evaluated algorithms, with results lower than 0.6 even with higher contamination.

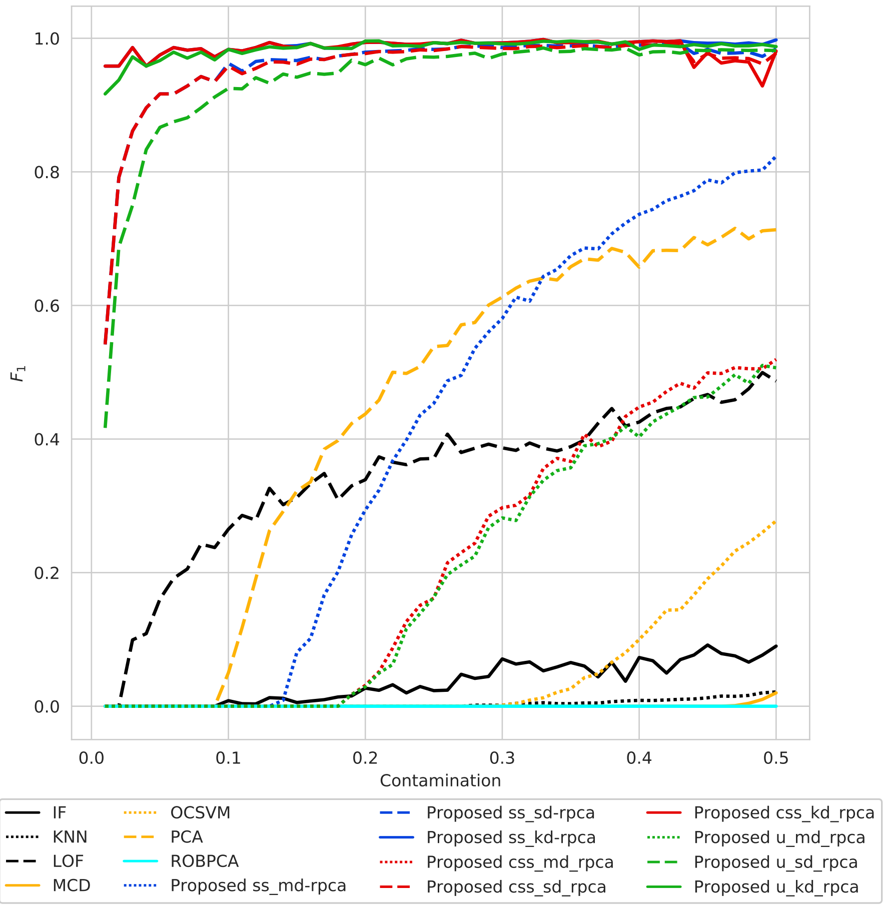


Figure 4.7 Anomaly detection on Lognormal distributed with Gaussian anomalies

The ss_md-RPCA and PCA algorithms perform similar, with worse detection rate for lower contamination and better performance with contamination higher than 0.4. However, the results of ss_md-RPCA and PCA are lower than all approaches of sd-RPCA and kd-RPCA. It is important to note that all mean-based approaches of m-RPCA presented lower results for anomaly detection on Lognormal data, in comparison to approaches based on skewness (sd-RPCA) and kurtosis (kd-RPCA), that achieved the best anomaly detection rates. However, the approaches of md-RPCA presented better results than IF, KNN, MCD, OCSVM and ROBPCA-AO.

The approaches based on kd-RPCA presented higher detection rate for all contamination, with scores near of 1.0. The semi-supervised and contaminated semi-supervised approaches for sd-RPCA performed similarly, but the unsupervised approach of sd-RPCA presented lower anomaly detection for lower contamination.

It is possible to note that the contaminated semi-supervised approaches of kd-RPCA and sd-RPCA performed similar to the semi-supervised approach of the same algorithms. These results highlight the resilience of the robust subspace learning even from contaminated training data for anomaly detection on Lognormal data.

Therefore, it is possible to observe that the semi-supervised approaches of m-RPCA overcome other approaches of m-RPCA and all the selected algorithms for anomaly detection on Lognormal distributed data with Gaussian anomalies.

Following we present the Tables 4.2, 4.3 and 4.4 to show the results of the selected algorithms and our proposals for anomaly detection on Gaussian, Pareto and Lognormal distributions, with 10%, 25% and 33% of contamination rate. The results are the F_1 for each scenario and algorithm, sorted by best result by algorithm for anomaly detection on Gaussian, Pareto and Lognormal distributions.

Table 4.2 Results for Simulated data set with 33% of contamination

Algorithm	**Gaussian + Uniform (F_1)**	**Pareto + Gaussian (F_1)**	**Lognormal + Gaussian (F_1)**
Proposed Semi-Supervised sd-RPCA	0.98	0.99	0.97
Proposed Semi-Supervised kd-RPCA	0.97	0.99	0.98
Proposed Contaminated Semi-Supervised sd-RPCA	0.98	0.94	0.96
Proposed Unsupervised sd-RPCA	0.98	0.82	0.93
Proposed Contaminated Semi-Supervised kd-RPCA	0.76	0.99	0.98
Proposed Unsupervised kd-RPCA	0.74	0.86	0.99
Proposed Semi-Supervised md-RPCA	0.98	0.90	0.66
PCA [121]	0.87	0.76	0.62
Proposed Contaminated Semi-Supervised md-RPCA	0.98	0.55	0.35
Proposed Unsupervised md-RPCA	0.98	0.54	0.31
Isolation Forest [125]	0.98	0.44	0.05
One-class SVM [122]	0.98	0.24	0.00
LOF [123]	0.35	0.40	0.36
KNN [124]	0.98	0.08	0.00
MCD [117]	0.98	0.02	0.00
ROBPCA-AO [31]	0.00	0.00	0.00

The results with 33% of contamination shows high scores of anomaly detection for Gaussian data with uniform anomalies by the evaluated algorithms, with exception to LOF and ROBPCA-AO. The semi-supervised approaches of m-RPCA presented the highest results for anomaly detection on Pareto data with Gaussian anomalies, and for anomaly detection on Lognormal data with Gaussian anomalies, as well as for Gaussian data with uniform anomalies.

The Table 4.3 also shows higher scores of semi-supervised m-RPCA for 25% of contamination, in comparison to the remain compared algorithms. It is important to note that the highest score for anomaly detection on Lognormal data with Gaussian contamination was the unsupervised kd-RPCA, with 0.99 F_1 score, and that the contaminated semi-supervised approaches of sd-RPCA and kd-RPCA presented scores near of the

Table 4.3 Results for Simulated data set with 25% of contamination

Algorithm	Gaussian + Uniform (F_1)	Pareto + Gaussian (F_1)	Lognormal + Gaussian (F_1)
Proposed Semi-Supervised kd-RPCA	0.96	1.00	0.98
Proposed Semi-Supervised sd-RPCA	0.97	0.99	0.96
Proposed Contaminated Semi-Supervised sd-RPCA	0.97	0.91	0.95
Proposed Unsupervised sd-RPCA	0.97	0.88	0.91
Proposed Contaminated Semi-Supervised kd-RPCA	0.63	1.00	0.97
Proposed Unsupervised kd-RPCA	0.62	0.97	0.99
Proposed Semi-Supervised md-RPCA	0.97	0.85	0.49
PCA [121]	0.91	0.75	0.56
MCD [117]	0.97	0.79	0.00
Proposed Contaminated Semi-Supervised md-RPCA	0.97	0.54	0.17
Proposed Unsupervised md-RPCA	0.97	0.53	0.14
Isolation Forest [125]	0.97	0.46	0.05
One-class SVM [122]	0.97	0.11	0.00
KNN [124]	0.97	0.05	0.00
LOF [123]	0.27	0.33	0.30
ROBPCA-AO [31]	0.00	0.00	0.00

results for semi-supervised approaches in scenarios with Gaussian, Pareto and Lognormal distributions.

The contamination rate of 10% shown by Table 4.4 shows worse results in comparison to contamination of 25% and 33%. However, the ROBPCA-AO presented high scores for anomaly detection on Gaussian data, overcoming LOF, css_kd-RPCA and u_kd-RPCA. ROBPCA also presented better results for anomaly detection on Pareto data in comparison to u_md-RPCA, css_md-RCAP, ss_md-RCAP, PCA, OCSVM, LOF, KNN and IF.

Table 4.4 Results for Simulated data set with 10% of contamination

Algorithm	Gaussian + Uniform (F_1)	Pareto + Gaussian (F_1)	Lognormal + Gaussian (F_1)
Proposed Semi-Supervised kd-RPCA	0.97	1.00	0.95
Proposed Semi-Supervised sd-RPCA	0.97	0.98	0.84
Proposed Contaminated Semi-Supervised sd-RPCA	0.96	0.90	0.82
Proposed Unsupervised sd-RPCA	0.97	0.93	0.70
Proposed Unsupervised kd-RPCA	0.50	0.97	0.98
Proposed Contaminated Semi-Supervised kd-RPCA	0.48	1.00	0.94
ROBPCA-AO [31]	0.97	0.87	0.00
PCA [121]	0.92	0.65	0.06
Proposed Semi-Supervised md-RPCA	0.97	0.59	0.00
Isolation Forest [125]	0.97	0.43	0.00
MCD [117]	0.97	0.40	0.00
Proposed Unsupervised md-RPCA	0.97	0.31	0.00
Proposed Contaminated Semi-Supervised md-RPCA	0.97	0.28	0.00
One-class SVM [122]	0.97	0.26	0.00
KNN [124]	0.97	0.01	0.00
LOF [123]	0.29	0.27	0.24

Taking into account the null hypothesis $H_2^{(N)}$ and the presented results, we can conclude that the robust subspace learning, adopted by md-RPCA, sd-RPCA and kd-RPCA, presented higher anomaly detection from imbalanced and skewed data than widely adopted algorithms for anomaly detection. Therefore, the presented results refute the null hypothesis $H_2^{(N)}$, which defines that the robust subspace learning does not improves the anomaly detection from imbalanced and skewed data.

4.6 Results of CTU-13 Experiment

In this section we present the experiment on network anomaly detection from the CTU-13 data set, evaluating the results of md-RPCA, sd-RPCA, kd-RPCA, Isolation Forest (IF) [125], k-Nearest Neighbors (KNN) [124], Local Outlier Factor (LOF) [123], Minimum Covariance Deteminant (MCD) [117], One-Class Support Vector Machines (OCSVM) [122], PCA [121] and ROBPCA-AO [31].

For this experiment we only evaluate the semi-supervised approach of md-RPCA, sd-RPCA and kd-RPCA, considering that the semi-supervised algorithms presented the best results on the simulated experiment and taking into account the observed resilience of the semi-supervised approach when the training data is contaminated. Additionally, the semi-supervised approach only requires the robust subspace learning for training, what can indicate less computational cost for network anomaly detection on new observations.

The CTU-13 data set is very imbalanced, with contamination rate between 0.06% and 8.86%. Therefore we adopted an uniform contamination rate of 33% for this experiment, considering that the simulated experiment showed better results for contamination higher than 30% for m-RPCA and for the selected anomaly detection algorithms.

Table 4.5 Network anomaly detection from CTU-13 with 33% of contamination

Algorithm	10	11	12	15	15-2	15-3	16	16-2	16-3	17	18	18-2	19	Avg
md-RPCA	0.83	0.79	0.95	0.78	0.78	0.87	0.95	0.87	0.80	0.82	0.83	0.82	0.51	0.81
kd-RPCA	0.76	0.76	0.90	0.82	0.57	0.76	0.91	0.50	0.80	0.73	0.83	0.81	0.48	0.74
sd-RPCA	0.25	0.75	0.34	0.64	0.50	0.75	0.86	0.50	0.77	0.33	0.82	0.81	0.21	0.57
PCA	0.33	0.64	0.69	0.65	0.55	0.62	0.75	0.50	0.77	0.33	0.82	0.01	0.61	0.56
MCD	0.18	0.29	0.09	0.34	0.79	0.62	0.04	0.58	0.20	0.41	0.20	0.20	0.36	0.33
IF	0.36	0.34	0.09	0.21	0.40	0.44	0.16	0.34	0.34	0.41	0.12	0.16	0.46	0.29
LOF	0.15	0.14	0.13	0.17	0.29	0.22	0.29	0.38	0.25	0.24	0.00	0.04	0.38	0.21
KNN	0.05	0.17	0.01	0.03	0.33	0.23	0.01	0.25	0.03	0.12	0.00	0.00	0.24	0.11
ROBPCA-AO	0.01	0.07	0.00	0.05	0.38	0.11	0.05	0.32	0.03	0.09	0.07	0.09	0.21	0.11

The Table 4.3 present the F_1 of each algorithm for all scenarios of CTU-13, and the last column presents the average F_1 of each algorithm for all scenarios.

It is possible to observe that md-RPCA, kd-RPCA and sd-RPCA overcome the remain algorithms in average results for all scenarios, according to the column **avg**. The sd-PRCA presented an average of 0.57 while md-RPCA obtained 0.81 and kd-RPCA 0.74 in average. The PCA algorithm performed similar to sd-RPCA in average, with results of

0.56 and 0.57, respectively. However the results of PCA and sd-RPCA for scenarios 12, 18-2 and 19 presented a large variation.

The md-RPCA algorithm presented an anomaly detection rate higher than 0.78 for almost all scenarios, with exception to scenario 19, where the anomaly detection rate of md-RPCA was 0.51. The anomalies of the scenario 19 are peer-to-peer botnet traffic generated by nsys.ay, which are related to botnet synchronization and not to network attacks, what can explain the low network detection rate of all evaluated algorithms, where the largest F_1 was 0.61 achieved by PCA.

The md-RPCA showed the best anomaly detection for 10 scenarios of a total with 13 scenarios. In the scenario 15 the best result was obtained by kd-RPCA, for the 15-2 scenario the best result was for MCD, and PCA was the best algorithm for scenario 19. Even thought md-RPCA not be the best result for scenarios 15, 15-2 and 19, the results of md-RPCA are very close to the best results for scenarios 15 and 15-2.

It is important to note that IF, KNN, LOF, MCD and ROBPCA-AO present the worse results for network attack detection on all scenarios of the CTU-13 data set, with average of 0.29, 0.11, 0.21, 0.33 and 0.11 respectively. From these algorithms, only MCD presented high result for one scenario, which was 0.79 for 15-2.

The CTU-13 data set is very challenging for anomaly detection approaches, due to the high imbalance and large volume of flows. However, CTU-13 is one of the up-to-date data sets for network attack detection and is one that provides the data imbalance observed in real anomaly detection problems. Unfortunately, was not possible to observe Pareto or Lognormal distributions on features of CTU-13, even though the finds of researches that highlight the fitting between these distributions and Internet traffic [29, 30].

The results of anomaly detection on CTU-13 reveals that m-RPCA algorithms obtain good results for network attack detection on contaminated data, overcoming widely adopted algorithms for outlier detection.

4.7 Conclusion

This work proposed the m-RPCA, which is approaches based on distances of moments computed from a robust subspace learned by RPCA, for anomaly detection on imbalanced and skewed data. We evaluated the anomaly detection rate of m-RPCA for simulated data and for the CTU-13 data set, which is composed by network traffic of botnets, attacks and background flows.

The m-RPCA can be divided into md-RPCA, sd-RPCA and kd-RPCA algorithms, to denote approaches of m-RPCA based on distances of mean, skewness and kurtosis, respectively. We also proposed to evaluate m-RPCA for semi-supervised, contaminated semi-supervised and unsupervised methods of anomaly detection.

The experimental results show that moment distances computed from robust subspace can improve the anomaly detection on skewed and imbalanced data set. The results also show that the m-RPCA can be adopted for network attack detection, with better results than widely adopted algorithms for outlier detection on the CTU-13 data set. Therefore, the observed results rejects the null hypothesis $H_1^{(N)}$ and confirms the alternative hypothesis $H_2^{(A)}$.

The results show that the semi-supervised approach for m-RPCA obtained better results than the contaminated semi-supervised and unsupervised approaches, and that m-RPCA presented good results even when the semi-supervised training was computed from contaminated data. This highlights the resilience of the robust subspace learning for the semi-supervised anomaly detection approach to deal with possible lack of known legitimate data for training.

The main contributions of this work were: the development and evaluation of novel approaches for anomaly detection on skewed and imbalanced data sets, by means of moments computed from a robust subspace learned by RPCA; the evaluation of the proposed approaches for anomaly detection on simulated data set and for network attack detection on real botnet traffic data set.

Future research can be directed to evaluate the application of the proposed approaches to different data sets and anomaly detection problems. The m-RPCA can also be extended to be online and to learn new subspace in an adaptive fashion, by means of new robust subspace learning from new observations or through robust subspace tracking.

In this work we adopt a multivariate analysis based on subspace learning from matrix decomposition, but a multidimensional approach can be evaluated in order to exploit the relationship between three or mode dimensions. Therefore, future works can evaluate the contribution of a robust tensor-based data modeling [70, 126], tensor decomposition [71] and multidimensional analysis for m-RPCA in problems of anomaly detection and network attack detection.

Future works can also consider to evaluate the sparse matrix $\boldsymbol{S}$ and its sparsity [127] for m-RPCA, in order to improve the accuracy and to be able to have element-wise anomaly detection.

5

Conclusion and Future Work

In the context of anomaly-based schemes, in thesis we propose propose a statistical approach based on signal processing techniques for detection of malicious traffic in computer networks. The proposed technique is based on eigenvalue analysis, model order selection (MOS) and eigen similarity analysis, where MOS and eigenvalue analysis are applied to detect time frames under attack. In addition, we evaluated the accuracy and performance of the proposed framework applied to an experimental scenario and to the DARPA 1998 data set.

We propose the m-RPCA, which an approach for anomaly detection on skewed and imbalanced data set, through distances of moments computed from a robust subspace learned by RPCA. The m-RPCA was evaluated for anomaly detection on simulated data and on the CTU-13 data set, which is composed by network traffic of botnets, network attacks and background flows.

Finally, it is proposed a proof of concept of an architecture to evaluate the user behavior analysis on offline mobile clients through Eigensimilarity and Model Order Selection (MOS), in order to implement an anomaly detection module based on user behavioral analysis.

This chapter is organized as follows. In Section 5.1, we discuss the conclusion remarks, in Section 5.2 it is present the main contributions and in 5.3 we propose opportunities for future work.

5.1 Conclusion

In order to be able to detect and avoid novel attacks and their variations, it is necessary to develop or improve techniques to achieve efficiency on resource consumption, processing capacity and response time. Moreover, it is crucial to obtain high detection accuracy and capacity to detect variations of malicious patterns. Recently, signal processing schemes have being applied to detect malicious traffic in computer networks, showing advances in network traffic analysis.

This thesis models the network traffic as a signal processing formulation for applying the Eigensimilarity, which is a framework for detection and identification of network attacks, that is based on eigenvalue analysis, model order selection (MOS) and eigen similarity analysis.

The Eigensimilarity was evaluated and the experimental results show that synflood, fraggle and port scan attacks can be detected accurately and with great detail in an automatic and blind fashion, applying signal processing concepts for traffic modeling and through approaches based on MOS and similarity analysis of learned subspace of eigenvalues and eigenvectors. The main contributions of Eigensimilarity were: the extension of an approach based on MOS combined with eigen analysis to blindly detect time frames under network attack; the proposal and evaluation of an eigen similarity based framework to identify details of network attacks, presenting accuracy of timely detection and identification of TCP/UDP ports under attack, as well as presenting acceptable complexity and performance regarding the processing time.

This thesis evaluated the effectiveness of MOS schemes for network attack detection, extending our previous work and showing that the analysis of the largest eigenvalues by time frames can be applied to detect the number of port scanning, and flood attacks, but still requiring more information for detailed attack detection. Therefore, we proposed a novel approach for detailed network attack detection, based on eigen similarity analysis.

Considering the offline mobile security context, an important issue faced by corporations that use cloud-based systems is how to provide security mechanisms to support offline corporate mobile clients. Aware of this problem and its importance, we proposed an architecture based on Eigensimilarity for behavioral anomaly detection in offline corporate mobile apps. As proof of concept, a fully working mobile application was developed to test the proposed security solution and acquired results provide evidence that besides achieving the desired security features, the solution also has positive results in terms of performance.

Anomalies can be hard to identify and separate from legitimate data due to the rare occurrences of anomalies in comparison to legitimate events, therefore anomaly detection algorithms have to be high discriminative, robust to contamination and able to deal with the imbalanced data problem.

This thesis has proposed and evaluated the m-RPCA, which can be divided into md-RPCA, sd-RPCA and kd-RPCA, that are algorithms based on moment distances from robust subspace for anomaly detection. These proposed algorithms were evaluated for semi-supervised, contaminated semi-supervised and unsupervised approaches. The experimental results show that moment distances computed fro robust subspace can improve the anomaly detection on skewed and imbalanced data set. The results also show that the m-RPCA can be adopted for network attack detection, with better results than widely adopted algorithms for outlier detection.

5.2 Contributions

We analyze problems related to detection of anomalies and information security issues, and propose new approaches to improve malicious behavior detection through signal processing techniques and subspace learning. The results of the work presented in this thesis provided the following contributions:

1. We proposed an approach based on eigen similarity analysis for extracting detailed information about accurate time and network ports under network attack, and evaluated the accuracy and performance of the proposed framework applied to an experimental scenario and to the DARPA 1998 data set;

2. We discussed the computational complexity of the proposed framework and evaluated the required processing time for tested scenarios;

3. We proposed an architecture and techniques for offline behavioral analysis of a corporate mobile client security architecture;

4. We discussed the processing time of the proposed framework for mobile devices;

5. We proposed approaches for network anomaly detection on skewed and imbalanced data set, through distances of moments computed from a robust subspace learned by RPCA;

5.3 Future Work

This thesis addresses some problems, but some problems are still open and others are emerging from current results. Thus, the following issues should be investigated as future work:

- Improvements for obtaining better false positive rates, as well as to make the Eigensimilarity able to identify sparse probe attacks or subtle behaviors, such as exfiltration or covert communication, considering the evaluation of a flow-based analysis and novel data sets;
- Evaluate the application of the proposed approaches to different anomaly detection problems, considering cases that are aware to behavioral analysis;
- Enhance the m-RPCA approaches to estimate the contamination or to not rely on previously defined contamination for anomaly detection;
- The m-RPCA can also be extended to be online and to learn new subspace in an adaptive fashion, by means of new robust subspace learning from new observations or through robust subspace tracking;
- Evaluate improvements given by a multidimensional modeling and adoption of tensor-based approaches for subspace learning, in order to analyze complex patterns that can be revealed by multidimensional analysis and tensor-based decomposition.

A

Critical Success Factor Analysis Based on Feature Selection

Critical Success Factor (CSF) is a management term for an element that is necessary for an organization or project to achieve its mission. CSFs represent the principal assets or areas that must be given investments to achieve better results. CSF analysis is one challenger strategic management tool, which can provide a robust and very practical assessment for strategic planners.

The identification of the most significant information for one problem is referred to as feature selection by the signal processing and data mining areas, as well as it can be formulated as a principal component problem, which is a widely adopted signal processing technique for data visualization and feature extraction. Feature selection aims to select a subset of relevant information from a larger data set, in order to improve: data visualization and data understanding, storage requirements, dimensionality, processing time, discriminative sensing, and to overcome overfitting problems to improve prediction and classification performance [128].

Recursive Feature Elimination (RFE) is a feature selection method for small sample classification problems. RFE seeks to improve generalization performance by recursively removing the least significant features whose deletion will have the least effect on training errors, according to the higher variance measured from the features [129].

We propose a critical factors analysis based on Principal Component Analysis (PCA) for visual analysis, and based on RFE combined with Support Vector Machine (SVM) [130] for classification and identification of critical success factors, applied to the survey that evaluates the IT governance of Brazilian public organizations, in order to identify the CSF for IT governance of the public sector according to TCU. Results show how PCA can help the data visualization and the discriminative visual analysis, and that SVM is the classifier that best performs on classification and obtains an accuracy of 91.42% to learn and classify according to TCU's IT governance evaluation of Brazilian public sector. Finally, SVM is used to highlight the more significant features identified by RFE, which are similar to CSFs previously identified by a qualitative analysis of the same data set.

This chapter is organized as follows. In Section A.1, related works are discussed. Section A.2 presents the data model and the evaluated data sets. Section A.3 describes the proposed approach for critical success factors analysis. Section A.4 discusses the experimental validation and presents the results, and Section A.5 draws the conclusions.

A.1 Related Works

Fink and Sukenik [131] explore the relationships among IT infrastructure capability and IT business value using PCA applied to all indicators of their study, resulting into 11 factors, with the first factor accounting for only 27.9% of the variance. This technique was used because the PCA extracts orthogonal factors that overcome the problem of multicollinearity.

Ramos *et al.* [4, 5] propose an overview regarding the evolution of scientific research on IT Governance critical success factors within the domain of public administration. By means of bibliometric analysis it was investigated seminal works regarding this theme, considering the characteristic key words found during our analysis. The results present 64 critical success factors with high impact on IT governance.

Guyon *et al.* [132] propose a method of gene selection utilizing Support Vector Machine methods based on Recursive Feature Elimination (RFE) and demonstrate that the selected genes yield better classification performance and are biologically relevant to cancer. The proposed method eliminates gene redundancy automatically and yields better and more compact gene subsets.

To the best of our knowledge, we are the first to propose a critical factors analysis based on PCA for visual discriminant analysis and based on RFE combined with SVM for CSF identification from IT governance data.

A.2 Data Model

The Brazilian Federal Court of Accounts (TCU, in Portuguese) surveys data regarding IT practices of Brazilian public organizations in order to audit IT governance. The data set with a consolidate view about the answers for this survey the IT governance index is called iGovTI. The iGovTI is composed by 201 multiple choice questions, used for ranking according to their IT governance, submitted to 349 organizations. The TCU computes the IT governance index and classifies the IT governance of each organization. Additionally, Ramos *et al.* [4] classifies each question regarding its relevance for IT governance through a qualitative analysis, and identify the CSFs for selected IT managers regarding IT governance.

A.3 An approach for Critical Success Factors Analysis

In this section we propose an approach for CSF Analysis based on visual discriminant analysis and based on feature selection, in order to identify the CSFs for IT governance according to iGovTI. Initially we conduct an analysis based on PCA to evaluate the relevance of each question according to their variance, and use the 2 most relevant features for a visualization of the iGovTI ranking. Furthermore, we propose a critical success factors analysis based on SVM for classification and based on RFE for identification of the most relevant factors.

A.3.1 Visual Discriminant Analysis based on PCA

PCA is a statistical technique commonly used for signal denoising, data compression, data visualization, feature extraction and dimensionality reduction, where a reduced number of features is extracted retaining as much information as possible [133]. It uses an orthogonal transformation to convert a set of correlated variables into a set of linearly uncorrelated variables, where the first principal components have the largest variance.

PCA is an orthogonal basis transformation into new basis, by diagonalizing the centered covariance matrix of a data set $\{\mathbf{x}_j \in \mathbb{R}^m, j = 1, ..., n\}$, defined by $\mathbf{C} = \frac{\mathbf{X}^\intercal \mathbf{X}}{n}$, where $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\intercal$ and the samples are assumed to have zero mean. The eigenvectors $\mathbf{v}_i$ of $\mathbf{C}$ are called principal components (PC), and the sample variance along $\mathbf{v}_i$ of $\mathbf{C}$ is given by the corresponding eigenvalue λ_i. Projecting onto the eigenvectors with the largest eigenvalues assumes that minimal information is lost, considering that in many applications these directions contain the most interesting information, such as in in data compression and denoising.

Initially, we compute the covariance matrix $\mathbf{C}$ of a zero mean data and visualize the data relationship for the sample covariance. Sample covariance is calculated by computing deviations of each measurement from the average of all measurements for that variable. Then the deviations for the two measurements are multiplied together separately for each subject. Finally these values are averaged. After that, the eigenvectors $\mathbf{v}_i$ and eigenvalues λ_i of $\mathbf{C}$ are computed through Singular Value Decomposition (SVD), then it is possible to evaluate the variance distribution of the extracted components through an empirical cumulative distribution function (ECDF). Evaluating the variance distribution we expect to visualize if some features concentrates the variance and indicates advantages for dimensionality reduction.

Finally, we propose to select the two features with the largest variance and evaluate the relationship between the two principal components and the iGovTI classification, in order to visualize if there is a segmentation according to the iGovTI ranking.

Considering that PCA combines attributes and creates new ones, with measurements from all of the original variables, it is hard to identify what original variables are most relevant. Therefore, we do not adopt PCA for classification and identification of CSF, thus it is still necessary an additional method to reveal what are the CSF for iGovTI. For this problem, we propose a CSF analysis based on RFE.

A.3.2 CSF analysis based on RFE

Feature selection aims the identification of the most significant information for one problem or algorithm, such as classification or prediction problems. RFE is a feature selection method for small sample classification problems by recursively removing the least significant features whose deletion will have the least effect on training errors, according to the higher variance measured from the features through a selected classifier.

Initially, it is necessary to identify one classifier that can identify the iGovTI classification of one organization according to a training data set. Therefore, we propose an algorithm evaluation in order to identify which one presents the best accuracy for iGovTI classification. The selected algorithm should be used by RFE to identify the CSF, that should be compared to results of the CSF qualitative analysis conducted by Ramos *et al.* [4].

Feature selection doesn't combine attributes, as PCA, but just evaluates their informative quality, predictive power and select the best set. Given an external estimator or classifier, that assigns weights to features, RFE is able to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coefficient attribute or through a feature importance attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached. RFE can also rank all features according to when they were eliminated.

The variables with the least effect on training errors or largest weights indicates more relevance for a classifier. Therefore, we propose to assume that the selected most relevant features are the CSF for iGovTI and also propose to validate this assumption against the results of the CSF qualitative analysis conducted by Ramos *et al.* [4].

A.4 Experiments and Results

In this section we present the experiments and results for the visual discriminant analysis based on PCA and for CSF analysis based on RFE, and discuss the results.

For the visual discriminant analysis we initially we compute the zero mean and the the covariance matrix $\mathbf{C}$ is computed from $\mathbf{X}$.The results are shown by Figure A.1.

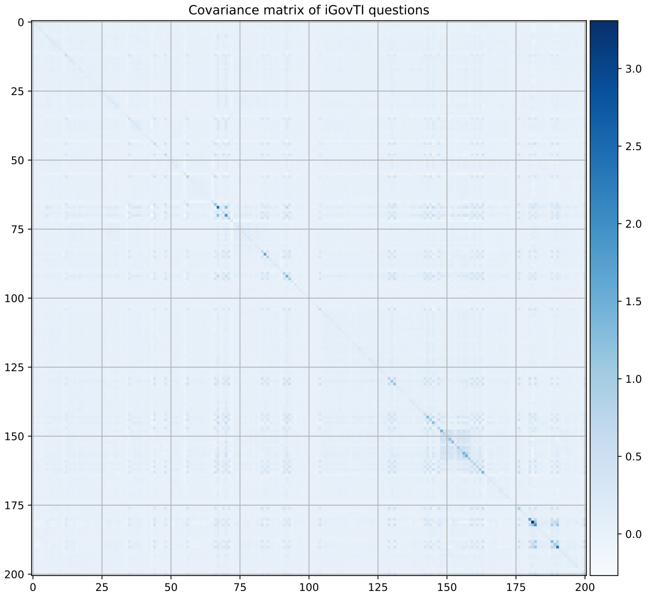


Figure A.1 Covariance matrix of iGovTI questions.

Figure A.1 presents the covariance matrix of 201 questions of iGovTI, where is possible to observe low covariance for the majority questions and high variance for just a few questions.

In the next step the eigenvectors $\mathbf{v}_i$ and eigenvalues λ_i of $\mathbf{C}$ are computed through SVD. Considering that the first principal components have the largest variance indicated by their eigenvalues, it is necessary to evaluate the variance distribution of the measured eigenvalues. Therefore, we present the Figure A.2, which shows the variance ECDF of the iGovTI questions and shows that 97% of the questions have variance lesser than 2, while just 3% of the questions have variance between 2 and 11. This result indicates that just a few principal components concentrates the most significant information and motivates the evaluation regarding the relationship between the principal components and the iGovTI classification.

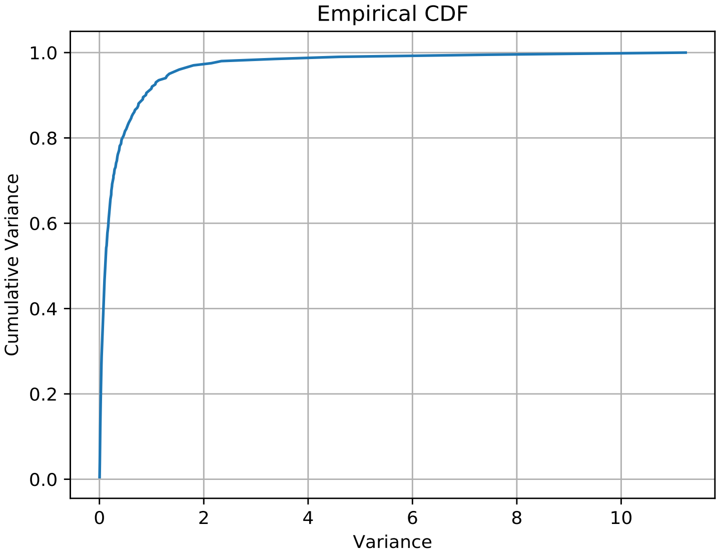


Figure A.2 Empirical CDF of variance.

Finally, we select the two largest eigenvalues and their correspondent components in order to evaluate the relationship between the iGovTI classification for 349 organizations and the two most informative variables. The Figure A.3 shows the scatter diagram that plots the organizations with higher iGovTI with larger circumferences and colors near of red, while organizations with lower iGovTI have colors near of blue and shorter circumferences. It is possible to observe that organizations with higher iGovTI are concentrated in the top left quadrant, with a visual separation of organizations with lower iGovTI, but without a clear distance separating the classes.

The visual discriminant analysis indicates that just a few principal components concentrates the most significant information and that the two principal components show a visual classification of organizations according to iGovTI classification, however it is not clear what are the CSF.

Hence, it is necessary additional techniques for CSF identification and we propose

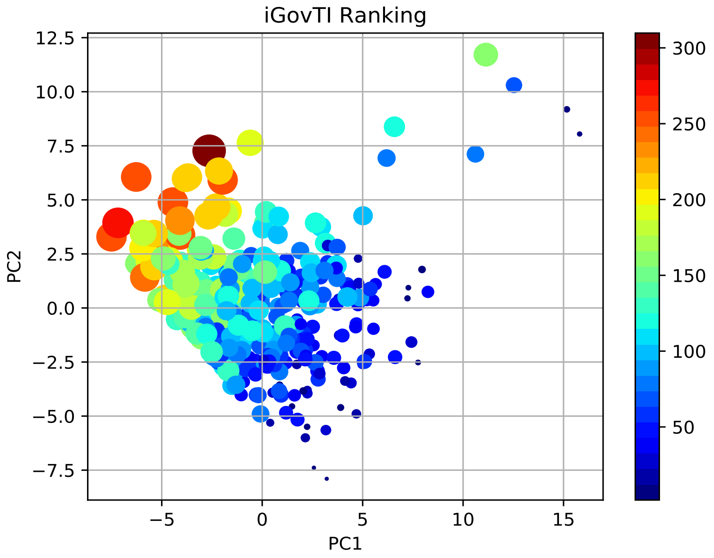


Figure A.3 iGovTI ranking from 2 principal components.

a critical success factors analysis based on RFE for identification of the most relevant factors for iGovTI.

RFE requires a classifier or predictor to evaluate its generalization performance by recursively removing the least significant features whose deletion will have the least effect on training errors, according to the higher variance measured from the features. We perform an algorithm evaluation in order to identify the classifier with the highest accuracy for iGovTI classification. The evaluated data set is composed of 201 questions (features) and 349 organizations (observations), where each organization is classified according to iGovTI classes, which are initial, intermediate and enhanced. We divide the data set into train and test, with the training data equivalent to 90% of the whole data set, while the test has 10 % of the whole data set.

The Table A.1 present the selected classification algorithms and the measured accuracy for iGovTI classification. According to results, SVM [130, 134] is the algorithm with highest classification accuracy, with 91.42%, while the second is Elastic Net [135], with 83.15% of accuracy. The results for iGovTI classification corroborates previous evaluations that highlight the advantages of SVM for classifications from small data sets [132]. Additionally, our results indicate that SVM is the best classifier for RFE, which corroborates the previous research of Guyon *et al.* [132], where is proposed SVM-RFE, that is a method of gene selection utilizing SVM methods based on RFE.

The next step is to use the selected algorithm as validator for recursive feature

Table A.1 Evaluation of classification accuracy for iGovTI

Algorithm	Mean Accuracy
Linear Regression [136]	0.3608
LDA [137]	0.6285
K-Nearest Neighbors [138]	0.7142
Linear SVM [139]	0.7142
Logistic Regression [140]	0.7714
SVM Regression [141]	0.8268
Lasso [142]	0.8274
Elastic Net [135]	0.8315
SVM [130]	0.9142

elimination in order to identify the CSF for iGovTI. Additionally, the RFE algorithm requires the number of desired most significant features. We adopt 54 for this variable, considering that is the number of CSF identified by Ramos *et al.* [4] in a qualitative analysis, that is the proposed target to validate the results of the CSF identification for iGovTI.

The RFE algorithm presents an ranking according to the significance of each feature, with an accuracy of 69,9% for CSF classification when compared to the CSF identified by Ramos *et al.* [4] in a qualitative analysis.

A.5 Conclusion

The proposed approach is evaluated and the experimental results show that the visual discriminant analysis, through PCA, highlights important characteristics of the data, and that a selected classifier and RFE can be applied for iGovTI classification and for CSF identification.

Results show how PCA can make the data discriminative, however it is hard to identify what original variables are most relevant. Therefore, it is still necessary an additional method to reveal what are the CSF for iGovTI. For this problem, we propose a CSF analysis based on RFE. Since RFE depends on a classifier to select the most relevant features, we perform an algorithm evaluation to identify the classifier with the highest performance, and results show that SVM [130] presents the best accuracy, with 91.42%.

Finally, SVM is used to select the more significant features identified by RFE, which are 69,9% similar to CSFs previously identified by a qualitative analysis of the same data set.

Bibliography (Own Publications)

[1] T. P. Vieira, D. F. Tenório, J. P. C. da Costa, E. P. de Freitas, G. Del Galdo, and R. T. de Sousa Júnior, “Model order selection and eigen similarity based framework for detection and identification of network attacks,” *Journal of Network and Computer Applications*, vol. 90, pp. 26–41, 2017.

[2] T. P. B. Vieira, J. P. C. L. da Costa, E. S. C. Vilaça, E. S. Gualberto, and R. T. de Sousa Júnior, “Moment distances from robust subspace for network attack detection,” *Journal of Network and Computer Applications*, To Appear.

[3] T. Galibus, T. P. d. B. Vieira, E. P. de Freitas, R. d. O. Albuquerque, R. T. de Sousa Júnior, V. Krasnoproshin, A. Zaleski, H. Vissia, G. del Galdo *et al.*, “Offline mode for corporate mobile client security architecture,” *Mobile Networks and Applications*, pp. 1–17, 2017.

[4] K. H. C. Ramos, R. T. de Sousa Junior, T. P. Vieira, and J. P. C. L. da Costa, “Discovering critical success factors for information technologies governance through bibliometric analysis of research publications in this domain,” *International Information Institute (Tokyo). Information*, vol. 19, no. 6B, p. 2193, 2016.

[5] K. H. C. Ramos, T. P. B. Vieira, J. P. C. L. da Costa, and R. T. de Sousa Júnior, “Multidimensional analysis of critical success factors for it governance within the brazilian federal public administration,” *The Light of External Auditing Data. 12th International CONTECSI*, 2015.

[6] T. Galibus, V. Krasnoproshin, T. P. Vieira, R. T. D. S. Júnior, J. P. C. Costa, E. P. De Freitas, A. Zaleski, and H. Vissia, “Organization of protection mechanisms for cloud storage services,” , no. 3 (97), 2016.

Bibliography

[7] A. Chuvakin, "2019 planning guide for security and risk management," 2019, accessed: 2019-06-15. [Online]. Available: https://blogs.gartner.com/anton-chuvakin/2018/10/30/2019-planning-guide-for-security-and-risk-management/

[8] C. of Economic Advisers, "The cost of malicious cyber activity to the u.s. economy," 2018, accessed: 2019-06-15. [Online]. Available: https://www.whitehouse.gov/articles/cea-report-cost-malicious-cyber-activity-u-s-economy/

[9] K. BISSELL, R. M. LASALLE, and P. D. CIN, "Ninth annual cost of cybercrime study," 2019, accessed: 2019-06-15. [Online]. Available: https://www.accenture.com/us-en/insights/security/cost-cybercrime-study

[10] A. Levi, "Can your big data company forego anomaly detection?" 2019, accessed: 2019-06-15. [Online]. Available: https://www.anodot.com/blog/big-data-anomaly-detection/

[11] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 217–228.

[12] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," 2008.

[13] S. Khattak, Z. Ahmed, A. A. Syed, and S. A. Khayam, "Botflex: A community-driven tool for botnet detection," *Journal of Network and Computer Applications*, vol. 58, pp. 144–154, 2015.

[14] F. Catucci, M. Isbitski, and R. Krikken, "Protecting web applications and apis from exploits and abuse," 2019, accessed: 2019-06-15. [Online]. Available: https://www.gartner.com/document/3907150

[15] J. Wang and I. C. Paschalidis, "Botnet detection based on anomaly and community detection," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 392–404, 2017.

[16] A. Wang, W. Chang, S. Chen, and A. Mohaisen, "Delving into internet ddos attacks by botnets: Characterization and analysis," *IEEE/ACM Trans.*

Netw., vol. 26, no. 6, pp. 2843–2855, Dec. 2018. [Online]. Available: https://doi.org/10.1109/TNET.2018.2874896

[17] P. Hevesi, "Ddos: A comparison of defense approaches," 2019, accessed: 2019-06-15. [Online]. Available: https://www.gartner.com/document/3907156

[18] D. Mudzingwa and R. Agrawal, "A study of methodologies used in intrusion detection and prevention systems (idps)," in *Southeastcon, 2012 Proceedings of IEEE.* IEEE, 2012, pp. 1–6.

[19] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[20] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[21] Accenture, "Gaining ground on the cyber attacker," 2018, accessed: 2019-06-15. [Online]. Available: https://www.accenture.com/pl-en/insights/security/2018-state-of-cyber-resilience-index

[22] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, 2019.

[23] L. Zonglin, H. Guangmin, Y. Xingmiao, and Y. Dan, "Detecting distributed network traffic anomaly with network-wide correlation analysis," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 2:1–2:11, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/752818

[24] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel pca-based network anomaly detection," in *2011 IEEE International Conference on Communications (ICC).* IEEE, 2011, pp. 1–5.

[25] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 4:1–4:16, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/837601

[26] N. Williams, “Featurespace raises £25m to drive international growth,” 2019, accessed: 2019-06-15. [Online]. Available: https://www.uktech.news/news/featurespace-raises-25m-to-drive-international-growth-20190128

[27] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008.

[28] H. He and E. Garcia, “Learning from imbalanced data,” *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[29] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 267–280.

[30] A. Leon-Garcia, “Probability, statistics, and random processes for electrical engineering,” 2017.

[31] M. Hubert, P. Rousseeuw, and T. Verdonck, “Robust pca for skewed data and its outlier map,” *Computational Statistics & Data Analysis*, vol. 53, no. 6, pp. 2264–2274, 2009.

[32] V. R. Basili, G. Caldiera, and H. D. Rombach, “The goal question metric approach,” in *Encyclopedia of Software Engineering*. Wiley, 1994.

[33] B. M. David, J. P. C. L. da Costa, A. C. Nascimento, D. Amaral, M. Holtz, and R. T. de Sousa Jr, “Blind automatic malicious activity detection in honeypot data,” in *The International Conference on Forensic Computer Science (ICoFCS)*, 2011.

[34] J. P. C. L. da Costa, E. P. de Freitas, B. M. David, A. M. R. Serrano, D. Amaral, and R. T. de Sousa Jr, “Improved blind automatic malicious activity detection in honeypot data,” in *The International Conference on Forensic Computer Science (ICoFCS)*, 2012.

[35] D. F. Tenório, J. P. C. L. da Costa, and R. T. de Sousa Jr, “Greatest eigenvalue time vector approach for blind detection of malicious traffic,” in *The International Conference on Forensic Computer Science (ICoFCS)*, 2013.

[36] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, “Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework,” *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[37] D. Networks, “Bad bot report 2019: The bot arms race continues,” 2019, accessed: 2019-06-15. [Online]. Available: https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019

[38] W. He, G. Hu, X. Yao, G. Kan, H. Wang, and H. Xiang, “Applying multiple time series data mining to large-scale network traffic analysis,” in *2008 IEEE Conference on Cybernetics and Intelligent Systems*, 2008, pp. 394–399.

[39] A. Ghourabi, T. Abbes, and A. Bouhoula, “Data analyzer based on data mining for honeypot router,” in *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*. IEEE, 2010, pp. 1–6.

[40] F. Raynal, Y. Berthier, P. Biondi, and D. Kaminsky, “Honeypot forensics,” in *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC*. IEEE, 2004, pp. 22–29.

[41] S. Almotairi, A. Clark, G. Mohay, and J. Zimmermann, “A technique for detecting new attacks in low-interaction honeypot traffic,” in *Internet Monitoring and Protection, 2009. ICIMP’09. Fourth International Conference on*. IEEE, 2009, pp. 7–13.

[42] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang, “Anomaly detection via online oversampling principal component analysis,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 7, pp. 1460–1470, July 2013.

[43] B. M. David, J. P. C. L. d. Costa, A. C. A. Nascimento, M. D. Holtz, D. M. Amaral, and R. T. d. Sousa Júnior, “A parallel approach to pca based malicious activitydetection in distributed honeypot data,” 2011.

[44] W. Z. A. Zakaria and M. L. M. Kiah, “A review on artificial intelligence techniques for developing intelligent honeypot,” in *Proceeding of: 8th International Conference on Computing Technology and Information Management, At Seoul, Korea*, 2012.

[45] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, “A multi-level intrusion detection method for abnormal network behaviors,” *Journal of Network and Computer Applications*, vol. 62, pp. 9–17, 2016.

[46] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, “A detailed analysis of the kdd cup 99 data set,” in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[47] Y. Chen, A. Wiesel, and A. O. Hero, “Robust shrinkage estimation of high-dimensional covariance matrices,” *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4097–4107, 2011.

[48] K. Yata and M. Aoshima, “Effective pca for high-dimension, low-sample-size data with singular value decomposition of cross data matrix,” *Journal of multivariate analysis*, vol. 101, no. 9, pp. 2060–2077, 2010.

[49] S. Jin and D. S. Yeung, “A covariance analysis model for ddos attack detection,” in *Communications, 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 1882–1886.

[50] J. P. C. L. da Costa, F. Roemer, D. Schulz, and R. T. de Sousa, “Subspace based multi-dimensional model order selection in colored noise scenarios,” in *2011 IEEE Information Theory Workshop*. IEEE, 2011, pp. 380–384.

[51] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa, “Multi-dimensional model order selection,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 26, 2011.

[52] J. Xiong, K. Liu, J. P. C. da Costa, and W.-Q. Wang, “Bayesian information criterion for multidimensional sinusoidal order selection,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3106–3110.

[53] J. P. C. L. da Costa, M. Haardt, F. Romer, and G. Del Galdo, “Enhanced model order estimation using higher-order arrays,” in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*. IEEE, 2007, pp. 412–416.

[54] J. P. C. L. da Costa, A. Thakre, F. Roemer, and M. Haardt, “Comparison of model order selection techniques for high-resolution parameter estimation algorithms,” in *Proc. 54th International Scientific Colloquium (IWK’09), Ilmenau, Germany*, 2009.

[55] J. Rajan and P. Rayner, “Model order selection for the singular value decomposition and the discrete karhunen–loeve transform using a bayesian approach,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 144, no. 2, pp. 116–123, 1997.

[56] H. Akaike, “A new look at the statistical model identification,” *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[57] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 2, pp. 387–392, 1985.

[58] A. Barron, J. Rissanen, and B. Yu, “The minimum description length principle in coding and modeling,” *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2743–2760, 1998.

[59] L. Zhao, P. Krishnaiah, and Z. Bai, “On detection of the number of signals in presence of white noise,” *Journal of Multivariate Analysis*, vol. 20, no. 1, pp. 1–25, 1986.

[60] M. O. Ulfarsson and V. Solo, “Rank selection in noist pca with sure and random matrix theory,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3317–3320.

[61] E. Radoi and A. Quinquis, “A new method for estimating the number of harmonic components in noise with application in high resolution radar,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 1177–1188, 2004.

[62] J. Grouffaud, P. Larzabal, and H. Clergeot, “Some properties of ordered eigenvalues of a wishart matrix: application in detection test and model order selection,” in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 5. IEEE, 1996, pp. 2463–2466.

[63] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt, “Model order selection for short data: An exponential fitting test (eft),” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2006.

[64] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical methods for rates and proportions*. John Wiley & Sons, 2013.

[65] T. Vieira, P. Soares, M. Machado, R. Assad, and V. Garcia, “Measuring distributed applications through mapreduce and traffic analysis,” in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*. IEEE, 2012, pp. 704–705.

[66] ——, “Evaluating performance of distributed systems with mapreduce and network traffic analysis,” in *2012 The Seventh International Conference on Software Engineering Advances - ICSEA*. IARIA, 2012, pp. 4–6.

[67] T. P. d. B. Vieira, S. F. d. L. Fernandes, and V. C. Garcia, “Evaluating mapreduce for profiling application traffic,” in *Proceedings of the First Edition Workshop on High Performance and Programmable Networking*, ser. HPPN ’13. New York, NY, USA: ACM, 2013, pp. 45–52. [Online]. Available: http://doi.acm.org/10.1145/2465839.2465846

[68] T. P. d. B. Vieira, “An approach for profiling distributed applications through network traffic analysis,” Master’s thesis, Universidade Federal de Pernambuco, 2013.

[69] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[70] J. P. C. da Costa, M. Haardt, and F. Romer, “Robust methods based on the hosvd for estimating the model order in parafac models,” in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*. IEEE, 2008, pp. 510–514.

[71] P. R. Gomes, J. P. C. da Costa, A. L. de Almeida, and R. T. de Sousa Jr, “Tensor-based multiple denoising via successive spatial smoothing, low-rank approximation and reconstruction for rd sensor array processing,” *Digital Signal Processing*, vol. 89, pp. 1–7, 2019.

[72] A. N. Khan, M. M. Kiah, M. Ali, S. Shamshirband *et al.*, “A cloud-manager-based re-encryption scheme for mobile users in cloud environment: a hybrid approach,” *Journal of Grid Computing*, vol. 13, no. 4, pp. 651–675, 2015.

[73] A. N. Khan, M. M. Kiah, M. Ali, S. A. Madani, S. Shamshirband *et al.*, “Bss: block-based sharing scheme for secure data storage services in mobile cloud environment,” *The Journal of Supercomputing*, vol. 70, no. 2, pp. 946–976, 2014.

[74] Yovel, Y, "Essential ways to protect my mobile apps," *Security Intelligence e-magazine*, 2014, accessed: 2019-06-15. [Online]. Available: https://securityintelligence.com/how-to-protect-mobile-apps-essentials/

[75] P. Quintiliano, J. P. C. L. da Costa, F. E. de Deus, and R. T. de Sousa Jr, "Computer forensic laboratory: Aims, functionalities, hardware and software," *ICoFCS 2013*, p. 72.

[76] A. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 393–413, 2014.

[77] A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278–1299, 2013.

[78] R. Mayrhofer, "An architecture for secure mobile devices," *Security and Communication Networks*, vol. 8, no. 10, pp. 1958–1970, 2015.

[79] H. Chang, A. Hari, S. Mukherjee, and T. Lakshman, "Design and architecture of a software defined proximity cloud," *Advances in Mobile Cloud Computing Systems*, p. 123, 2015.

[80] Y. Xia, Y. Liu, C. Tan, M. Ma, H. Guan, B. Zang, and H. Chen, "Tinman: eliminating confidential mobile data exposure with security oriented offloading," in *Proceedings of the Tenth European Conference on Computer Systems*. ACM, 2015, p. 27.

[81] P. Kulkarni and R. Khanai, "Addressing mobile cloud computing security issues: a survey," in *Communications and Signal Processing (ICCSP), 2015 International Conference on*. IEEE, 2015, pp. 1463–1467.

[82] D. M. Shila, W. Shen, Y. Cheng, X. Tian, and X. S. Shen, "Amcloud: Toward a secure autonomic mobile ad hoc cloud computing system," *to appear*, 2016.

[83] M. Heydari, S. M. S. Sadough, M. S. Farash, S. A. Chaudhry, and K. Mahmood, "An efficient password-based authenticated key exchange protocol with provable security for mobile client–client networks," *Wireless Personal Communications*, vol. 88, no. 2, pp. 337–356, 2016.

[84] G. Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang, “Trusted data sharing over untrusted cloud storage providers,” in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 97–103.

[85] J. Yang, H. Wang, J. Wang, C. Tan, and D. Yu, “Provable data possession of resource-constrained mobile devices in cloud computing,” *Journal of networks*, vol. 6, no. 7, pp. 1033–1040, 2011.

[86] W. Itani, A. Kayssi, and A. Chehab, “Energy-efficient incremental integrity for securing storage in mobile cloud computing,” in *Energy Aware Computing (ICEAC), 2010 International Conference on*. IEEE, 2010, pp. 1–2.

[87] W. Ren, L. Yu, R. Gao, and F. Xiong, “Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing,” *Tsinghua Science & Technology*, vol. 16, no. 5, pp. 520–528, 2011.

[88] W. Lu and A. A. Ghorbani, “Network anomaly detection based on wavelet analysis,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 4, 2009.

[89] D. M. Lyra-Leite, J. P. C. L. da Costa, and J. L. A. de Carvalho, “Improved mri reconstruction and denoising using svd-based low-rank approximation,” in *2012 Workshop on Engineering Applications*. IEEE, 2012, pp. 1–6.

[90] McAfee, “Mcafee labs threats report,” 2019, accessed: 2019-06-15. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-dec-2018.pdf

[91] Kaspersky, “Mobile cyber threats,” *Kaspersky Lab & INTERPOL Joint Report*, 2014, accessed: 2019-06-15. [Online]. Available: http://media.kaspersky.com/pdf/Kaspersky-Lab-KSN-Report-mobile-cyberthreats-web.pdf

[92] , “Storgrid protected cloud storage security whitepaper,” 2016, accessed: 2016-01-15. [Online]. Available: http://www.storgrid.com

[93] J. C. Dos Anjos, M. D. Assunção, J. Bez, C. Geyer, E. P. De Freitas, A. Carissimi, J. P. C. Costa, G. Fedak, F. Freitag, V. Markl *et al.*, “Smart: An application framework for real time big data analysis on heterogeneous cloud environments,” in *2015 IEEE International Conference on Computer and Information Technology;*

Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE, 2015, pp. 199–206.

[94] J. C. d. Anjos, T. Galibus, C. F. Geyer, G. Fedak, J. P. C. Costa, R. Pereira, and E. P. de Freitas, "Fast-sec: an approach to secure big data processing in the cloud," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 34, no. 3, pp. 272–287, 2019.

[95] D. Networks, "Bad bot report 2019: The bot arms race continues," 2019, accessed: 2019-06-15. [Online]. Available: https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019

[96] C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: Classification of skewed data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 50–59, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007738

[97] C. Pascoal, M. R. De Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust pca for internet traffic anomaly detection," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 1755–1763.

[98] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 665–674.

[99] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *computers & security*, vol. 45, pp. 100–123, 2014.

[100] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden, "Robpca: a new approach to robust principal component analysis," *Technometrics*, vol. 47, no. 1, pp. 64–79, 2005.

[101] D. Acarali, M. Rajarajan, N. Komninos, and I. Herwono, "Survey of approaches and features for the identification of http-based botnet traffic," *Journal of Network and Computer Applications*, vol. 76, pp. 1–15, 2016.

[102] G. Gu, P. A. Porras, V. Yegneswaran, M. W. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation." in *USENIX Security Symposium*, vol. 7, 2007, pp. 1–16.

[103] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1. ACM, 2007, pp. 109–120.

[104] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[105] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE signal processing magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[106] G. Lerman and T. Maunu, "An overview of robust subspace recovery," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1380–1410, 2018.

[107] Y. Cherapanamjeri, P. Jain, and P. Netrapalli, "Thresholding based efficient outlier robust pca," *arXiv preprint arXiv:1702.05571*, 2017.

[108] "Rad: Time series anomaly detection," https://github.com/Netflix/Surus/, 2018, accessed: 2019-06-15.

[109] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[110] S. Garcıa, "Identifying, modeling and detecting botnet behaviors in the network," *Unpublished doctoral dissertation, Universidad Nacional del Centro de la Provincia de Buenos Aires*, 2014.

[111] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in neural information processing systems*, 2009, pp. 2080–2088.

[112] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[113] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *preprint*, vol. 12, p. 2, 2009.

[114] P. M. G. Reis, J. P. C. da Costa, R. K. Miranda, and G. Del Galdo, "Audio authentication using the kurtosis of esprit based enf estimates," in *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2016, pp. 1–6.

[115] D. Zwillinger and S. Kokoska, *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.

[116] P. J. Rousseeuw, “Least median of squares regression,” *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[117] P. J. Rousseeuw and K. V. Driessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[118] Y. Zhao, Z. Nasrullah, and Z. Li, “Pyod: A python toolbox for scalable outlier detection,” *arXiv preprint arXiv:1901.01588*, 2019.

[119] S. Arlot, A. Celisse *et al.*, “A survey of cross-validation procedures for model selection,” *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[120] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” 2011.

[121] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, “A novel anomaly detection scheme based on principal component classifier,” MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, Tech. Rep., 2003.

[122] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[123] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[124] F. Angiulli and C. Pizzuti, “Fast outlier detection in high dimensional spaces,” in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2002, pp. 15–27.

[125] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[126] P. R. Gomes, A. L. de Almeida, J. P. C. da Costa, and G. Del Galdo, “Tensor-based methods for blind spatial signature estimation under arbitrary and unknown source covariance structure,” *Digital Signal Processing*, vol. 62, pp. 197–210, 2017.

[127] K. Liu, F. Roemer, J. P. C. da Costa, J. Xiong, Y.-S. Yan, W.-Q. Wang, and G. Del Galdo, "Tensor-based sparsity order estimation for big data applications," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 648–652.

[128] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[129] X.-w. Chen and J. C. Jeong, "Enhanced recursive feature elimination," in *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*. IEEE, 2007, pp. 429–435.

[130] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[131] L. Fink and E. Sukenik, "The effect of organizational factors on the business value of it: universalistic, contingency, and configurational predictions," *Information Systems Management*, vol. 28, no. 4, pp. 304–320, 2011.

[132] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[133] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal component analysis*. Springer, 1986, pp. 115–128.

[134] P. M. G. I. Reis, J. P. C. L. da Costa, R. K. Miranda, and G. Del Galdo, "Esprit-hilbert-based audio tampering detection with svm classifier for forensic analysis via electrical network frequency," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 853–864, 2016.

[135] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[136] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 2014.

[137] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[138] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE transactions on computers*, vol. 100, no. 7, pp. 750–753, 1975.

[139] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[140] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[141] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[142] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.