Multivariate Statistical Network Monitoring for Network Security based on Principal Component Analysis

Noemí Marta Fuentes García

Supervisors: José Camacho Paéz
Gabriel Maciá Fernández

Department of Signal Theory, Telematics and Communications
Universidad de Granada

This dissertation is submitted for the degree of
PhD
Program in Information and Communication Technologies

Editor: Universidad de Granada. Tesis Doctorales
Autor: Noemí Marta Fuentes García
ISBN: 978-84-1306-823-7
URI: http://hdl.handle.net/10481/67941

CITIC-UGR December 2019

A mis padres y a Antonio.
Os quiero.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Noemí Marta Fuentes García
December 2019

Agradecimientos

Estos últimos cuatro años han sido probablemente los más intensos de mi vida. Hacer una tesis doctoral no es duro, es lo siguiente, no solamente por el trabajo en sí que conlleva (que también), sino por la montaña rusa de emociones que supone llevar a cabo esta hazaña. Por eso quisiera agradecer a todos los que habéis estado ahí de un modo u otro apoyándome o haciendo posible que esto saliera adelante. ¡Gracias!

Me gustaría comenzar por Pepe y Gabri, mis dos directores, sin vosotros dos esto no habría sido posible. Gracias por haber confiado en mí desde el principio y haberme guiado y apoyado durante este tiempo. Me ha encantado trabajar con vosotros. Gracias también a todo mi grupo de investigación por acogerme con los brazos abiertos y hacer que me sintiera integrada como una más. Gracias a José María, mi supervisor durante mi estancia en Ámsterdam. Gracias por darme la oportunidad de trabajar en Shell contigo y por ayudarme en mis inicios en un país que para mí era extraño.

Gracias a mis compañeros de despacho, porque han hecho que mi estancia en el CITIC fuera mucho más que eso. Gracias a Paco, Roberto, Pablo, Gabri, Ignacio, Fermín y Diego. Roberto, fuiste el primero que conocí y el primero que abandonó el nido. Gracias por haberme presentado a Pilar y haber hecho todo lo posible porque me sintiera arropada. Gabri, nos conocíamos desde nuestros inicios en la carrera y coincidimos tras el paso de los años realizando nuestras tesis. Pablo, gracias por tus consejos y tus conversaciones de despacho. Paco, gracias por tus consejos de "visual thinking" y por confiar

en mí para que te asesorara al respecto. Gracias por haberme ayudado siempre y por esas largas charlas que hacían más amenos los días.

Gracias a todos los que habéis hecho que olvide la tesis por momentos. A María A., por haber compartido conmigo parte de mis días en Ámsterdam y haberme invitado a sus clases de yoga. Hiciste que la estancia fuera más amena y gané una amiga. A Rocío, que desde que comencé la carrera ha estado a mi lado. Por animarme a salir los domingos y aceptar que no todos se puede tomar café cuando se hace una tesis. A María W. porque empezó siendo mi profesora de yoga y acabó siendo más que eso. Gracias a Paco y a Sandra, por esos momentos de juegos en los que para todos lo más importante era ganar al "Shushi Go" o al "Kwatro". Gracias por las escapadas culinarias.

Por último, y no por ello menos importante, gracias a mi familia por haberme apoyado siempre. Gracias mamá y papá por haberme dado todo sin pedir nada a cambio, por haberme dedicado los días y las horas. Gracias por haber creído siempre en mí y por animarme a "dar lo mejor posible". Sin vosotros no habría llegado hasta aquí. Gracias a mi hermano y a Laura, por estar siempre ahí aunque no podamos vernos tanto como quisiéramos. Gracias por vuestra comprensión. Gracias Antonio por formar parte de mi vida. Gracias por ser mi luz incluso en los días más oscuros. Gracias por estar a mi lado, por apoyarme y darme ánimo. Los dos hemos pasado por esto juntos y los dos lo hemos conseguido juntos.

Abstract

Currently we live in hyper-connected world, which is one of the main causes for the fast propagation of Information Technology (IT) Security attacks. An IT Security incident can impact both in the economy and the reputation of the organization that suffers it. Thus, IT Security is a prior concern for any organization. Another important issue related to IT Security threats is that the time required for compromising a network is, on average, in the order of minutes, while the security team may need months to detect an incident after it takes place. This makes it necessary to enhance the mechanisms of intrusion detection to improve the capability of prioritization and classification of IT security alarms. With the appropriate tools, the security team can detect the incidents timely without being overwhelmed by an excessive number of alarms.

Network security is of utmost importance within IT Security, and it aims to make the communications infrastructure secure from the point of view of the IT. In general, there are three approaches for network security: *prevention*, *detection* and *response*. These approaches can be combined to achieve a comprehensive security system. A practical combination of the detection and response dimensions is the so-called Network Security Monitoring (NSM), which is an approach that aims to detect the incidents in a network by monitoring the network traffic. NSM is carried out by collecting, combining and analyzing different sources of information, in order to detect and notify intrusions. There are two main techniques for incident detection: *Signature-*

based, which allows to detect attacks from previously defined patterns; and *Anomaly-based*, which allows to detect deviations from the normal behavior in a network, captured in a previously trained model.

Multivariate Statistical Network Monitoring (MSNM) is an NSM methodology that follows an anomaly-based detection scheme that extends the Multivariate Statistical Process Control (MSPC) theory, developed in the area of industrial process research. MSPC consists in two phases: phase I, detection of assignable causes of variation in the calibration data that are corrected and eliminated until the process is under Normal Operation Condition (NOC); and phase II, monitoring of new data to detect (and diagnose) anomalies. MSNM applies this philosophy to traffic network data, adding two prior steps: parsing and fusion, which are needed to combine information from different data sources in NSM. MSNM is useful to prioritize and diagnose anomalies, which is congruent with the security team's workflow.

In this PhD, we start from the MSNM methodology and introduce a number of enhancements: *i)* a pre-processing method to consider the cyclostationarity of the data (*e.g.* the cycles existing during day and night or weeks and weekends), *ii)* a methodology for the comparison of diagnosis methods, and *iii)* a univariate method for diagnosis. Furthermore, the pre-processing and diagnosis methods, as well as some of other existing extensions for MSNM are evaluated and compared with other reference methods using a real network data set for the first time. The application on real network data allows to assess the MSNM extensions under realistic conditions, yielding a more accurate perspective of their performance.

This research work shows the existing symbiosis between industrial processes and network security, introducing enhancements that are of interest for both topics and that open new lines of research exploring the synergy between MSPC and MSNM.

Table of contents

List of figures xvii

List of tables xxiii

1 Introduction 1

1.1 Motivation . . . 3

1.1.1 The Cost of IT (In)Security . . . 3

1.1.2 Network Security . . . 7

1.2 Objectives . . . 12

1.3 Main Contributions . . . 12

1.3.1 Articles . . . 15

1.3.2 Conference Papers . . . 16

1.4 Organization of this Thesis . . . 18

2 Network Security Monitoring 21

2.1 Components of a Network Security Monitoring System . . . 24

2.1.1 Sensors . . . 25

2.1.2 Integrators . . . 34

2.2 Solutions for Network Security Monitoring . . . 41

2.2.1 Intrusion Detection Systems (IDSs) . . . 42

2.2.2 Integrators . . . 43

2.2.3 Tool Collections . . . 46

I Multivariate Statistical Monitoring **49**

3 Multivariate Statistical Process Control **51**
3.1 Principal Component Analysis 54
3.2 Anomaly Detection 55
3.3 Diagnosis 58
3.3.1 The *Smearing* Problem 59
3.4 Batch MSPC (BMSPC) 59
3.4.1 Batch Monitoring Cycle 61
3.4.2 The Parameter Stability Problem 64

4 Multivariate Statistical Network Monitoring **67**
4.1 MSNM 69
4.1.1 Parsing 70
4.1.2 Fusion 71
4.1.3 Detection 72
4.1.4 Diagnosis 74
4.2 Extensions on the MSNM application 74
4.2.1 Extensions in the Fusion Step 76
4.2.2 Extensions in the Detection Step 78
4.2.3 Extensions in the Diagnosis Step 82
4.2.4 Extensions for Big Data 82

II Materials **87**

5 Materials and Methods **89**
5.1 Implementation Tools 90
5.1.1 Multivariate Exploratory Data Analysis (MEDA)-Toolbox 91
5.1.2 MultiVariate Batch (MVBatch) Toolbox 94
5.2 Data Generation 95

5.2.1 The UGR'16 Dataset 97
5.2.2 Virtual Network 98
5.2.3 *Saccharomyces cerevisiae* Simulator 98
5.2.4 Synthetic Data 100

III Contribution to the Multivariate Statistical Network Monitoring 101

6 Pre-processing 103
6.1 State-of-the-art Pre-processing Methods 107
6.1.1 The Parameter Stability Problem in the Pre-Processing Context 108
6.2 Ratio number-of-Observations-to-the-number-of-Parameters (ROP) Enhancement Alternatives 110
6.3 PARAmeters from More Observations (PARAMO) 113
6.3.1 *Uniform PARAMO (U-PARAMO)* 113
6.3.2 *eXponential PARAMO (X-PARAMO)* 114
6.3.3 Configuration Values for PARAMO 116
6.4 RAw DAta Filtering (RADAF) 117
6.5 Oversights on the Application of ROP Enhancement Approaches 119
6.5.1 Negative Effects of Asymmetric Windows 119
6.5.2 Negative Effects of RADAF 121
6.6 Materials and Methods 122
6.6.1 Process Control: *Saccharomyces Cerevisiae* 123
6.6.2 Metrics for Evaluation of the pre-processing proposals 127
6.6.3 Finding Comparable Configurations for the Exponential and Uniform Moving Window Methods 129
6.7 Evaluation of the Pre-processing Proposal 130
6.7.1 Results of the Main Experiment 130
6.7.2 Results applying RADAF in model building 139

6.8 Conclusions 143

7 Diagnosis **145**
7.1 State-of-the-art Diagnosis Methods 147
7.1.1 Contribution Plots (CP) 147
7.1.2 Reconstruction-Based Contributions (RBC) 148
7.1.3 The observation-based Missing-data method for Exploratory Data Analysis (oMEDA) 149
7.2 Univariate Squared: a Different Approach for Diagnosis . . 152
7.3 Methodology for Comparison of Diagnosis Methods 153
7.3.1 Step 1. Generation of Anomalies with Known Diagnosis 154
7.3.2 Step 2. Definition of a metric to evaluate the diagnosis performance 156
7.3.3 Step 3. Experimental Design 157
7.4 Evaluation of Diagnosis Methods 159
7.4.1 Case of Study I: Simulated Synthetic Data 160
7.4.2 Case of Study II: Simulated Communication Network Traffic 164
7.4.3 Case of Study III: Data Set for Process Control . . . 169
7.5 Conclusions 177

8 MSNM Extensions Applied to Real Data **179**
8.1 MSNM extensions 182
8.2 Materials and Methods 184
8.2.1 Anomaly Detection Assessment 184
8.2.2 UGR'16 dataset 185
8.2.3 MSNM application 191
8.3 Results for Standard MSNM 199
8.4 Results for Hierarchical MSNM 206
8.5 Comparison of Hierarchical and Standard Approaches 211

8.6 Conclusions 212

IV Conclusions 215

9 Conclusions 217

9.1 Conclusions 220

9.2 Future Work 223

References 225

Appendix A List of Acronyms 247

Appendix B Related Terms 253

Appendix C Oversights on the Application of RBC for the D-statistic 259

Appendix D Efficiency Calculation for Standard and Hierarchical Fusion 263

Appendix E Resumen amplio en castellano 265

List of figures

1.1 Data breaches in 2018 and 2019. 5
1.2 Network Security approaches and examples grouped by main goal. 8

2.1 Network Security Monitoring based in the general model of (Cyber)security Management [16, 19]. 23
2.2 Workflow of data through an NSM system. 24
2.3 Example of the general scheme for an NSM system. 26
2.4 Modes of detection engines. 38

3.1 D- and Q-statistics and their corresponding theoretical control limits. 57
3.2 Example of diagnosis applying oMEDA, obtained using the ***MEDA-Toolbox*** [43]. 59
3.3 Visual representation of the first variable of the *Saccharomyces Cerevisiae* process simulated using ***MVBatch*** [89] toolbox without applying any synchronization or alignment, and after applying the multi-synchro algorithm [89, 91]. 62
3.4 Main types of unfolding into a single matrix. Figure adapted from [46, 52]. 63

4.1 Example of raw data source for MSNM: `NetFlow` record from the UGR'16 [130]. 70
4.2 Counters corresponding to the example in Fig. 4.1. 71
4.3 Conceptual representation of the fusion step. 72
4.4 Example of detection using the Tscore. 73
4.5 Example of diagnosis using oMEDA. 75
4.6 MSNM steps and the main extensions. 75
4.7 Examples of hierarchical MSNM. 79
4.8 Different arrangements of the data (2-way vs 3-way). 80
4.9 Additional steps for MSNM to deal with Big Data. 83
4.10 oMEDA diagnosis for the first anomalous observation of the UGR'16. 84
4.11 Summary on the raw data after De-parsing an anomalous and a NOC observation. 85

5.1 Example of functionalities of the MEDA-Toolbox. 92
5.2 Example of Big Data functionalities of the MEDA-Toolbox. 93
5.3 MEDA-Toolbox Graphical User Interface (GUI). 94
5.4 Example of functionalities across the GUI of the MVBatch Toolbox. 96
5.5 Connections with destination `HTTPS` port from UGR'16. . . 99

6.1 Visual representation of Trajectory Centering and Scaling (TCS) and Variable Centering and Scaling (VCS) pre-processing. 107
6.2 Trajectories applying different pre-processing methods for the Specific Oxygen Uptake Rate of the *Saccharomyces Cerevisiae* data set. 111
6.3 Graphical representation of window-based pre-processing. . 112
6.4 Effect of applying different levels smoothing on X-PARAMO. 117

6.5 Comparison of the pre-processing parameters computed with TCS and X-PARAMO based on non-symmetric and symmetric windows. 120
6.6 Comparison of the pre-processing parameters after applying RADAF with the symmetric exponential law and TCS. . . . 122
6.7 Synthetic faults generated from the simulated fermentation process of the *Saccharomyces cerevisiae* cultivation. 125
6.8 Homogenization of the configuration values of X-PARAMO to the reference method, U-PARAMO. 131
6.9 Comparison and interactions between enhancing approaches and settings for the standard deviations and the loadings. . . 134
6.10 Comparison between settings under study for the means. . . 134
6.11 Comparison between moving window methods for the standard deviations. 135
6.12 Comparison and interactions between the 3 factors for FaultII. 135
6.13 Comparison and interaction between enhancing approach and moving window method for Faults I and III. 136
6.14 Comparison between type of pre-processing under study for the means and standard deviations. 137
6.15 Comparison between type of pre-processing under study for Faults I, II and III. 138
6.16 Comparison and interactions between enhancing approach and settings under study for the loadings. 139
6.17 Comparison between the approach of enhancing and TCS for the loadings. 140
6.18 Comparison and interactions between the tree factors for Faults I, II and III. 141
6.19 Comparison between TCS, PARAMO and RADAF for Faults I, II and III. 142

7.1 Example of grouping observations applying oMEDA. 151
7.2 Example of diagnosis applying oMEDA. 152
7.3 Least Significant Difference (LSD) plots for Thin, Square, and Fat matrices. 162
7.4 Percentage of anomalies generated on each statistic for 1 Principal Component (PC) and for the number of PCs that captures 75% of the total variance. 163
7.5 Ratios for 1 PC and Thin, Square and Fat matrices simulated with SimuleMV. 165
7.6 Ratios for 2 PCs and T, 4 PCs and S, and 11 PCs and F simulated with SimuleMV 166
7.7 Percentage of anomalies generated on each statistic for 1 PC and 2 PCs corresponding to the *Communications Network Traffic* data set. 168
7.8 LSD plots for Network Traffic Data. 169
7.9 Ratios in the Q-statistic for 1 and for 2 PCs corresponding to the *Communications Network Traffic* data set 170
7.10 Percentage of anomalies generated on each statistic for 1 PC and 2 PCs for (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) corresponding to the *Saccharomyces cerevisiae* process simulation. 172
7.11 LSD plots for matrices Thin, Square, and Fat matrices corresponding to the *Saccharomyces cerevisiae* process simulation. 173
7.12 Ratios for 1 PC for Thin, Square and Fat matrices corresponding to the *Saccharomyces cerevisiae* process simulation. . . . 175
7.13 Ratios for 2 PCs for Thin, Square and Fat matrices corresponding to the *Saccharomyces cerevisiae* process simulation. 176

8.1 Standard fusion: Aggregating (**A**-fusion) and Concatenating (**C**-fusion). 183

8.2 Comparison in terms of TCP SYN counts between a day with a DoS attack (red) and an attack-free day (blue). 187
8.3 IRC traffic in UGR'16 during June. 189
8.4 Wrong diagnosis of the botnet (model calibration includes anomalous IRC traffic in June). 190
8.5 Statistics by IP and port (output from the nfdump tool). . . . 191
8.6 Standard topology for **C**-fusion and **A**-fusion. 194
8.7 Hierarchical topologies (a) **H1**, (b) **H2**, (c) **H3**, and (d) **H4**. . 197
8.8 Receiver Operating Characteristics (ROC) curves for the standard fusion **C** and **A**. 200
8.9 Area Under the Curve (AUC) grouped by attack for the standard fusion **C** and **A**. 201
8.10 Univariate Squared (U-Squared) diagnosis applying **C**-fusion for DoS, Scan11, Scan44 and Botnet. 202
8.11 Sorted U-Squared diagnosis applying **C**-fusion for Scan11 and Scan44. 203
8.12 ROC curves for the hierarchical fusion. 207
8.13 AUC grouped by attack for the hierarchical fusion. 207
8.14 Hierarchical topologies **H4b** and **H4c**. 208
8.15 Comparison between hierarchical fusions, including H4b and H4c for ROC curves AUC per individual attack. 209
8.16 Comparison between standard and hierarchical fusions for ROC curves AUC per individual attack. 214

List of tables

1.1 Main objectives (MO). 13
1.2 Secondary Objectives (SO). 13

6.1 Synthetic datasets generated with the fermentation process of the *Saccharomyces Cerevisiae* cultivation. 124
6.2 Factors, levels and interactions considered for the 3-factor ANalysis Of VAriance (ANOVA) studies. 132
6.3 Results of the 3-Factor ANOVAs for the parameter stability evaluation. 132
6.4 Results of the 3-Factor ANOVAs for the monitoring performance evaluation. 135
6.5 Levels for the 1-factor ANOVA studies. 137

7.1 Parameters involved in the Monte Carlo Simulation. 160
7.2 Parameters involved in the verification using *Network data*. . 167
7.3 Parameters involved in verification using *Saccharomyces cerevisiae* process data. 169

8.1 Criteria to assign end-points to virtual routers (with sensors). 188
8.2 Distribution of the attacks in the virtual routers (with sensors). 189

8.3 Configuration settings for X-PARAMO. W represents the total size of the window (given in minutes), and λ corresponds to the forgetting factor. 195
8.4 Diagnosis for **A**-fusion applying U-Squared and oMEDA. . . 204
8.5 Diagnosis for **C**-fusion applying U-Squared and oMEDA. . . 205
8.6 Diagnosis for the hierarchical fusion applying U-Squared at the top layer. 210
8.7 Diagnosis for the hierarchical fusion applying U-Squared at the second layer. When this diagnosis does not apply, it is signaled with '-'. 211
8.8 Diagnosis for the leaf routers applying U-Squared. When this diagnosis does not apply, it is signaled with '-'. The '*' denotes that this is not considered an incorrect diagnosis, since we previously verified that, indeed there exists anomalous `telnet` traffic in the background. 211

9.1 Main objectives (MO). 219
9.2 Secondary Objectives (SO). 220

1

Introduction

"If I would know what I am doing, I wouldn't call it research, would I?"
Albert Einstein, Nobel Prize in Physics in 1921

"To start, press any key. Where's the ANY key?"
Homer, The Simpsons

Contents

1.1 Motivation 3
1.1.1 The Cost of IT (In)Security 3
1.1.2 Network Security 7
1.2 Objectives 12
1.3 Main Contributions 12
1.3.1 Articles 15
1.3.2 Conference Papers 16
1.4 Organization of this Thesis 18

The term **Information Technology (IT) Security** refers to the set of techniques, rules and standards that allow to maintain safe and secure any IT system and communications network [82]. A term which is related to IT Security is that of *information security*. According to many authors [16, 82, 218], **Information security** should meet at least the following requirements: *1) Confidentiality*, *2) Integrity*, and *3) Availability*. **Confidentiality** consists in preventing non-authorized users from accessing or reading restricted information. **Integrity** consists in avoiding data to be altered by any unauthorized part. **Availability** guarantees that a given service or data is ready for its use. Most authors also add *4) Accountability, which* encompasses *Non-repudiation*, *Authentication* and *Authorization* (also called *Access Control*). **Accountability** is the capability of identifying who is the responsible for an event occurrence in the system, what caused such event, and when did it take place. This implies **Non-repudiation**, which ensures that none of the parties involved in an event can reject that they took part in such event; **Authentication**, which ensures that each of the parties involved in an event are identified; and **Authorization**, which ensures that only the parties granted with access rights will actually

access a given service or resource [16, 82, 218]. Other authors also include *Privacy*, which has increased its relevance in the last years with the expansion of the application of Machine Learning and Big Data techniques [137], as well as the trend of being permanently connected [197]. **Privacy** is closely related to Confidentiality, since both refer to the information protection. The main difference is that Privacy is the *right* of any natural or legal person to not sharing their personal information with anyone [146, 167, 210]; while Confidentiality refers to the *agreement* of avoiding non-authorized users to access any restricted information [146, 167, 197, 210].

1.1 Motivation

An **IT security incident** happens when the IT Security of an organization or company is compromised due to any reason, violating any of the IT Security requirements enumerated before. The number of IT security incidents has increased over the last years. In 2014, eighteen thousand IT Security incidents were detected by the Instituto Nacional de Ciberseguridad (INCIBE) in Spain. By 2018, the number had increased to more than one hundred thousand [99].

1.1.1 The Cost of IT (In)Security

In the *V Jornadas Nacionales de Investigación en Ciberseguridad* (JNIC, 2019), the **Chief Information Security Officer (CISO)** in Telefónica shared an interesting reflection: "*The question is not «Can I suffer a cyber attack?» but «When am I going to be (cyber) attacked*?» or «*How many cyber attacks I am going to suffer?*»". He also emphasized that the key is prevention and, over all, the availability of a contingency plan [172]. The high potential of the technology at the moment, in combination with the high degree of interconnection of the society, makes the balance to be usually on the attackers' side. This implies both a challenge and an opportunity at the same time,

especially for the development of IT Security research, as well as for the related industries. Nowadays, the IT Security market involves more than 85,000M €/year, which is expected to increase to 190,000M €/year by 2023. In addition, between 2021 and 2027 the European Union is expected to invest more than 2,100M € in IT Security [99]. According to the Gartner report [83], this year (2019) the world investment in information security will be greater than 110,000M€.

A relevant part of the IT Security incidents are *data breaches*[1]. Fig. 1.1 shows the data breaches in the last two years colored by data sensitivity [136]. In 2018 it was revealed that 50M of user profiles were gathered from Facebook without permission of their owners [25]. More recently, in September of 2019, a public database was disclosed with more than 400M of phone numbers including, in some cases, other private data, such as name or gender [96, 136]. This might cost a fine of billions of euros to Facebook [18, 78, 176, 196]. A technical error in Twitter made the password of 300,000M users to be stored in plain (readable) text. The company solved and published the error in 2018, and asked the affected users to change their password [77]. This bug will also probably imply a billionaire fine to the company [176]. Data breaches not only affect big companies, since "*no organization is too large or too small to fall victim to a data breach*" ("Data Breach Investigations Report" from 2019, Verizon) [206]. The cost of a data leak for a Small and Medium-sized Enterprise (SME), such as a private clinic or a small store, with only one hundred clients, might reach the thousand of euros in a year. A data breach has an average impact of 126€/year per record and person [136, 183, 199, 205]. There are many causes for a data breach, such as

[1] A **data breach** is a type of security incident that occurs when someone accesses and extracts personal or confidential information without any authorization [183, 199].

an unauthorized access to a system, a *malware*[2] infection, or insufficient data protection [14, 104, 183, 199].

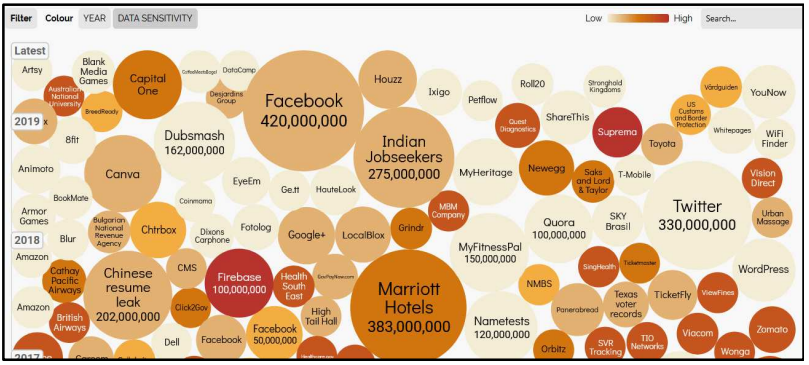


Fig. 1.1 Data breaches in 2018 and 2019, colored according to the data sensitivity. Size of nodes reflect the number of affected users. Source: Information is beautiful [136].

According to the Verizon report in 2019 [206], the most affected industry by IT Security incidents is the *Public Administration*, which has registered more than twenty thousand cases, two thirds parts related to espionage. In the rest of the industrial sectors, these incidents are mostly motivated by an economical reward. Some of the most affected industries are *Information*, *Financial and Insurance*, *Healthcare*, and *Professional, Technical and Scientific Services*. These incidents are, in general, caused by errors in Web Applications or Privilege Misuse, and involve the compromise of personal data. The *threats*[3] are commonly external, meaning that the actors do not belong to the organization target of the attack. However, there are some sectors, such as *Healthcare*, *Information* or *Educational Services*, where the threats are almost

[2] **Malware** refers to any type of software that is built with the aim of damaging any device (*e.g.* computer or telephone). It is used with different purposes, such as stealing information or denying access to legitimate users [15, 133].

[3] A **threat** is any situation or event that may damage a system or network. This refers to unveiling, destroying, modifying or denying the access to the data or services [16]. A threat that comes true is termed an **attack**.

50% internal, meaning that the actors are people belonging to the company or organization [206].

In addition, given the hyper connected world we live in today, an attack can be easily propagated, causing a great impact, especially in the economy and the reputation of any company or organization. Recently, the company Varonis presented the study "60 Must-Know Cybersecurity Statistics for 2019" [205], which unveils some alarming figures about IT Security: "*More than* 41% *of the companies have more than one thousand of sensitive files (including credit cards and health care records) unprotected*". "*By 2021, the cost of the damage related with the cybercrime is projected to hit six trillion of dollars annually*". Thus, cybersecurity and IT Security are some of the main concerns for leaders in any organization.

Another important problem when facing up IT Security threats is that the average time required to compromise a system or a network is relatively small (in the order of minutes) in comparison to the average time that security people need to detect the incident after it takes place (which may range from days to months). Thus, it is really important to work on the reduction of this time gap for detection and response. On the other hand, new attacks are developed every day and they evolve quickly [84]. For this reason, security detection mechanisms need to incorporate approaches to detect new attack strategies, previously unseen. Finally, the security team usually receives more alarms than they can handle [9, 62, 135]. In this sense, it is desirable that the mechanisms for detection allow the adequate *prioritization*[4] of the alarms. Applying prioritization in *IT Security events*[5] enhances the efficiency

[4]The **prioritization** of events (also known as events **triage**) is based on one or more criteria. These criteria may be, for example, the impact or the magnitude of the event.

[5]An **IT Security event** in a system or network refers to any undesired situation or modification in the system or network that occurs for a period of time and that is susceptible to be detected by the security system. If the event is detected, this usually generates an **alert**, which is recorded as an individual log as a part of a file or database.

in anomaly detection, since it helps the security team to focus in solving the most important alarms first.

1.1.2 Network Security

Network security is of utmost importance within IT Security. The objective of network security is to make the communications infrastructure accomplish all the security requirements (Confidentiality, Integrity, Availability, Accountability, Non-repudiation, Authentication, Authorization and Privacy) both in the independent hosts and in the network as a whole. There exist different standpoints to address network security. These aspects can be classified in: *prevention*, *detection* and *response*. These are not exclusive; on the contrary, they are usually applied together to achieve a more secure network infrastructure [16, 19, 82]:

- **Prevention** refers to the actions (including software deployment and user awareness) taken by an organization to prevent attacks from being successful.
- **Detection** is the part of the security system that implements mechanisms to identify potential attacks, mainly by monitoring the network.
- **Response** allows the system or network to react once an attack is detected [82].

Although a high percentage of the efforts are still focused on preventing attacks, the solutions and techniques based in detection and response are gaining more and more relevance. This makes sense if we consider the reflection made in [20]: "*prevention is similar to have a sheepdog, which guards hundreds of sheep, and faces a pack of wolves; while detection is to be able to find out a wolf hiding in a flock of sheep, and response is to act when the wolf is discovered*". There is the general acceptance in the IT

Security community that, sooner or later, prevention measures are surpassed by attackers. At that point, detection and response mechanisms need to be applied.

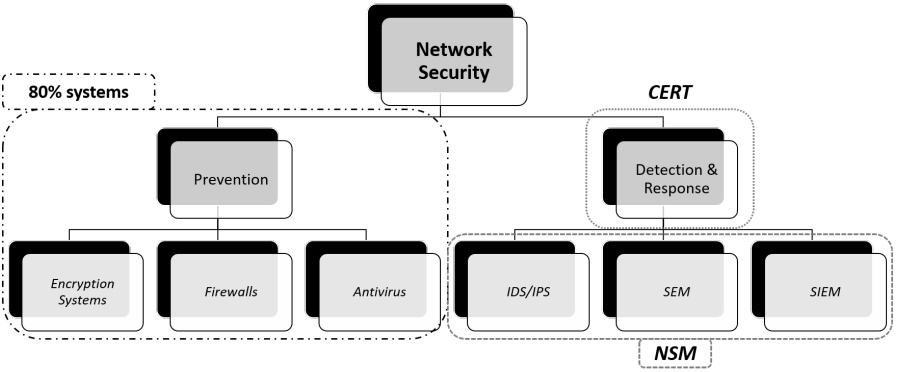


Fig. 1.2 Network Security approaches and examples grouped by main goal.

Fig. 1.2 shows the classification of the aforementioned security aspects, providing some examples of solutions that can be enclosed on each of them. When we talk about **prevention**, the most basic tool of protection is a **firewall**, which filters the communications coming in and out to a network (or host), and provides access control. The **antivirus** software scans the files in the hosts looking for known patterns (signatures) of malware and, if there is a match, it handles the file properly. Finally, a main mechanism of prevention is **data encryption**, which preserves the confidentiality and integrity in the communications and may also ensure the authentication and non-repudiation.

The **Computer Emergency Response Teams (CERTs)** are groups of security specialists that aim to detect and respond to cybersecurity incidents, warning and/or advising the rest of the citizens and organizations about them. CERTs are typically dependant either on governmental organizations or private big companies [72, 184]. The Computer Security Incident Response

Teams (CSIRTs) are frequently considered to be equivalent to the CERTs. However, they are usually more focused in detection and response, rather than in prevention [169, 185]. A list with the European CERT and CSIRT entities can be found in [73].

Some of the most extended tools for **detection** (and usually also for **response**) are introduced in the following paragraphs.

Intrusion Detection Systems (IDSs) are a set of techniques to detect suspicious activity (possible intrusions) by monitoring and analyzing the events in a network or a device [62, 103, 218]. When these systems also allow to deploy defensive responses to the attacks, they are called **Intrusion Prevention Systems (IPSs)**. These responses can be carried out by discarding or modifying the packets related to the attack [103]. Some of the IDSs have evolved to **Security Event Management (SEM)** systems, which allow the *correlation*[6] of events from different sources, improving the detection capability [62].

Security Information and Event Management (SIEM) systems are the combination of two different systems. On the one hand, a **Security Information Management (SIM)** system allows regulatory compliance, the analysis and notification of the events, as well as long-time storage of such events. This makes it possible to perform forensic analysis once an attack has taken place. On the other hand, the SEM allows the real-time monitoring and correlation of events. Thus, the objective of a SIEM system is to aggregate and analyze the information collected from a number of *sensors* to detect, select, classify and validate incidents in a network [112]. In addition, a SIEM system generates reports for the compliance of security policies, useful to pass audits. Finally, another important feature is that SIEM systems allow the visualization and

[6] In the context of IT Security, the term **correlation** means finding connections among distinct data sources or IT Security events, rather than being used with the traditional statistical meaning.

prioritization of the events, thus helping *security operators*[7] to interpret and understand the alarms [87, 103].

Finally, **Network Security Monitoring (NSM)** is an approach intended to detect the attacks in a network by monitoring the network traffic [134, 218]. This is carried out by collecting, correlating and analyzing traffic, to detect intrusions, with the aim of alerting and notifying such intrusions [20, 171]. Sometimes, NSM also implies responses or actions when an attack is detected. SIEM systems may be seen as examples of NSM systems [19, 20, 32].

In general terms, the incident detection process can be classified into *Signature-based* and *Anomaly-based* [68, 81, 103, 105]. The former identifies attacks from previously defined patterns. The latter detects deviations from the normal behavior in a network or system in relation to a previously trained model. Signature-based systems cannot detect *zero-day*[9] attacks while anomaly-based are theoretically able to do it. On the contrary, anomaly-based approaches tend to generate a high number of (false) alarms. Thus, one of the main challenges for detection is to reach a balance between both the capability to detect *zero-day* and the reduction of false alarms [62, 218].

NSM Methodologies

One of the main problems in NSM (as previously pointed out in the more general context of IT Security) is the high time of response of the security

[7] A **security operator** is a person in charge of administering and monitoring the security system in an organization, while a **security analyst** is in charge of analyzing and discovering *vulnerabilities* and *risks* in the organization. Both security operators and analysts are usually interchangeable terms, and they are people who take part of the **security team**.

- A **vulnerability** is an *asset*[8] that could lead to its unauthorized exploitation. A vulnerability may exist due to a bad design, implementation or even for intentional reasons [16, 19].
- A **risk** is the probability of suffering any damage or lost. Its value can be considered a combination of the threat, vulnerability and relevance of the asset [19, 82].

[9] A ***zero-day*** attack is an attack that had not been previously seen and, thus, its features and signature are not known.

team in comparison to the time needed to compromise a system. This makes it necessary to prioritize the IT Security events, so that security operators can optimize their working protocols and manage the alarms in a more effective manner [9, 62, 84, 135]. For example, *neural networks* are used for anomaly detection but the results are hard to interpret and, thus, prioritization of the alarms is limited. In the last years, the multivariate analysis has also being explored for anomaly detection in the context of IT Security [42, 118]. One of the best known multivariate methods is *Principal Component Analysis (PCA)*. Furthermore, the multivariate analysis can be combined with other techniques, such as Exploratory Data Analysis [42] or visualization tools, allowing the triage and interpretation of alarms. In addition, the multivariate analysis offers an advantage over most Machine Learning (ML) techniques: the *diagnosis*. The **diagnosis** ease the identification and understanding of the root causes of a given incident, thus helping in the triage of the alarms [36]. Although diagnosis in the context of black box ML exists [147], this is still in its infancy.

Multivariate statistical approaches based on PCA were proposed at the beginning of the 2000s for intrusion detection [110, 118]. One of the main advantages of PCA is its unsupervised nature, which does not require the prior specification of potential anomalies in the system. For this reason PCA is a powerful tool to build systems that detect both known and new types of anomalies, which is of principal importance for network security practitioners in order to detect *zero day* attacks. The use of PCA to build detection engines in the context of NSM was proposed by Lakhina *et. al* in a pioneering work in the early 2000's [118]. The main goal of the proposal is to distinguish normal from abnormal network traffic by means of a PCA model, and to diagnose the root causes of anomalies. There have been several modifications of the original PCA-based approach [26, 27, 66, 119, 168]. However, most of these proposals still maintain part of the drawbacks identified in the original work [118]. This motivated the development of the *Multivariate Statistical Network Monitoring (MSNM)* methodology, which leverages the similarities

between network traffic and industrial processes and inherits the procedures in the well-established Multivariate Statistical Process Control (MSPC). In MSNM, first it is needed to perform feature extraction and data fusion, which allow dealing with different data sources; and then the MSPC theory based in PCA is employed. This connection between network and process data allows us to take advantage of the existing solutions in the process control industry for intrusion detection, yielding a perfect cooperation between both areas [42].

This PhD is focused on anomaly detection based systems in the context of both the NSM and the process monitoring. More precisely, the aim of this work is to push the recent developments on Multivariate Data Analysis in MSNM [42] and propose enhancements of these techniques that contribute both to MSNM and MSPC. For this reason, this thesis is conducted using data both from industrial processes (*Saccharomyces Cerevisiae* cultivation process) and network traffic data, as well as synthetic data.

1.2 Objectives

The objective of this PhD is to deal with the main research problems related to the detection and diagnosis of incidents in network security by applying multivariate data analysis. To carry out this general goal, the individual objectives are defined in Table 1.1.

Some additional specific objectives are also defined as a part of the research plan for this PhD, which are shown in Table 1.2.

1.3 Main Contributions

The following paragraphs summarize the main contributions of this PhD and relate them to the Objectives defined in Section 1.2.

Objective	Description
MO1	To design methods or algorithms for anomaly detection based in multivariate analysis. These methods should reduce the number of false alarms and allow the detection of *zero-day* attacks.
MO2	To design methods or algorithms for the accurate diagnosis of anomalies.

Table 1.1 Main objectives (MO).

Objective	Description
SO1	To evaluate the proposed techniques for anomaly detection and diagnosis, which implies the comparison with the state-of-the-art methods.
SO2	To apply the proposed techniques to real network data.

Table 1.2 Secondary Objectives (SO).

Cyclo-stationary Pre-processing. A pre-processing method for cyclo-stationary data was proposed as a result of a short research stay in Amsterdam. The stay took place in **Shell Global Solutions International B.V.**, under the supervision of Dr. José María González-Martínez. The goal of the stay was to study Multivariate Anomaly Detection techniques for *cyclo-stationary* data. The data are **cyclo-stationary** when we can find cycles in their behavior. Examples of cyclo-stationary data are traffic network and industrial batch processes [52, 91, 148–150]. As a result of the stay, a new alternative for pre-processing cyclo-stationary data was developed. This is useful to improve the detection capabilities of multivariate anomaly detection methods. The proposed method reduces the uncertainty in the pre-processing parameters in comparison to the reference method in the literature. The proposed pre-processing method enhances the anomaly detection, which is one of the main

objectives of the thesis (**MO1**). This method was also evaluated and compared with the reference method with simulated and real data (**SO1** and **SO2**).

Diagnosis of Anomalies. A methodology for the comparison of diagnosis methods was designed during the first year of the PhD. This is a comprehensive methodology that allows to consider the main factors that affect the diagnosis within the framework of Experimental Design and ANalysis Of VAriance (ANOVA). The methodology is developed to provide low uncertainty results, thanks to the application of the Monte Carlo approach. On the other hand, a univariate diagnosis method was proposed. This method aims to solve one of the main problems in the multivariate diagnosis methods, the *smearing* effect. This effect is a consequence of the correlation between the variables, which can make an anomalous variable to contaminate non-affected variables. The proposal of this diagnosis method pursues one of the main goals of the thesis (**MO2**). Using the proposed methodology for comparison, the univariate diagnosis method was evaluated and compared with the state-of-the-art diagnosis methods with simulated and real data (**SO1** and **SO2**).

Evaluation of Multivariate Anomaly Detection Methodologies in the Context of NSM to Real Network Data. In 2015 the Multivariate Statistical Network Monitoring (MSNM) was proposed as a methodology for multivariate anomaly detection for NSM. Since then, there have been a number of variants [33, 36, 40, 129, 192, 195], some of them developed in the context of this PhD.

As a part of the contribution of this research work, some of the MSNM variants were applied to real network data collected from an Internet Server Provider (ISP) (**SO2**). A new comparison of the pre-processing [192] and the diagnosis proposals [195] against the state-of-the-art methods was carried out (**SO1**). For the pre-processing, the comparison was carried out in terms of the performance of anomaly detection, defined by the relation between the number

of false positives and false negatives. For the diagnosis, the comparison was carried out in terms of accuracy of the variables signaled as anomalous, and the simplicity of visual interpretation.

One of the features of NSM is that it allows the combination and correlation of data from different data sources. The combination of the different data sources to provide a single data matrix can be performed in different ways. The hierarchical combination of the data was proposed as an MSNM extension in [129]. This type of fusion implies the definition of several levels and the building of a different model in each of the defined levels. The hierarchical fusion was also evaluated with real data (**SO2**). For the hierarchical approach, several scenarios were defined to cover distinct cases of study.

1.3.1 Articles

The results of this PhD have been published in different research journals with high impact factor. These contributions are detailed below:

- **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2018). Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control. *Chemometrics and Intelligent Laboratory Systems*, 172:194–210

- Camacho, J., Maciá-Fernández, G., **Fuentes-García, N. M.**, and Saccenti, E. (2017b). Semi-supervised multivariate statistical network monitoring for learning security threats. *Transactions on Information Forensics and Security*, 14(8):2179–2189

- Camacho, J., García-Giménez, J. M., **Fuentes-García, N. M.**, and Maciá-Fernández, G. (2019b). Multivariate Big Data Analysis for Intrusion Detection: 5 steps from the haystack to the needle. *Computers and Security (COSE)*, 87

- **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019b). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. *Journal of Chemometrics*, 33(11)

1.3.2 Conference Papers

The results of this PhD have been shared with the research community in different conferences, both international and national, which are detailed below:

International Conferences

- **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016c). Fault Diagnosis: Contribution plots vs oMEDA. In *XVI Chemometrics in Analytical Chemistry (CAC), Barcelona (Spain)*

- **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2017b). A Univariate Approach for Diagnosis in PCA-MSPC. In *Scandinavian Symposium on Chemometrics (SSC15), Naantali (Finland)*

- González-Martínez, J. M., **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2017). Parameter stability and its effects on bilinear modelling of batch processes. In *Mini Arctic Workshop, Valencia (Spain)*

- **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019c). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. In *Scandinavian Symposium on Chemometrics (SSC16), Oslo (Norway)*

National Conferences

- **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016a). Comparación de métodos de diagnóstico de anomalías en monitorización estadística multivariante de redes. In *Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Menorca (Spain)*

- Magán-Carrión, R., Camacho, J., Maciá-Fernández, G., and **Fuentes-García, N. M.** (2017). Esquema Jerárquico de Monitorización y Detección de Anomalías en Red: Aplicación Práctica. In *III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), Madrid (Spain)*

- Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., **Fuentes-García, N. M.**, García-Teodoro, P., and Therón-Sánchez, R. (2017). UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red. In *XIII Jornadas de Ingenieria Telematica (JITEL2017), Valencia (Spain)*, pages 71–78

- Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., **Fuentes-García, N. M.**, García-Teodoro, P., and Therón Sánchez, R. (2018). Un resumen de: UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red basados en cicloestacionariedad. In *IV Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), San Sebastián (Spain)*, pages 117–118

- **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2019a). Evaluación de mejoras en la monitorización estadística multivariante para la detección de anomalías en tráfico ciclo-estacionario. In *V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), Cáceres (Spain).*

Outreach Sessions

- **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016b). Diagnóstico de Anomalías: Gráficos de Contribución vs oMEDA. In *I Jornadas de Investigadores en Formación Fomentando la interdisciplinariedad (JIFFI), Granada (Spain)*

- **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2017a). Defending the network. Detection and Diagnosis of Anomalies. In *CITIC-Coffees, Granada (Spain)*

1.4 Organization of this Thesis

This thesis is organized in five parts plus appendices. Each part is composed of several chapters. The first part is devoted to the Introduction and is organized in two chapters: This present chapter introduces the motivations and objectives of this work, while Chapter 2 describes the state-of-the-art in Network Security Monitoring.

The second part is focused on the Multivariate Statistical Monitoring, which is the basis of this work. The MSPC methodology, developed in the process industry, is introduced in Chapter 3. The MSNM (which is based in the MSPC extension for NSM) is detailed in Chapter 4.

The third part of the thesis presents the Materials and Methods used for this research work, including software, metrics and main data sets (Chapter 5).

The Contributions to the Multivariate Statistical Network Monitoring of this PhD work are presented in the fourth part, which consists of three chapters: Chapter 6, where an alternative approach to enhance the pre-processing in MSNM/MSPC applied to three-way tensors is proposed, and Chapter 7, where a novel diagnosis method in MSNM/MSPC is proposed. These contributions, together with some of the extensions for the MSNM, are applied on a real network data set in Chapter 8.

Finally, the fifth part includes Chapter 9, which presents the Conclusions of this thesis.

2

Network Security Monitoring

"Trust everyone and not trust anyone are two vices: But in the one there is more virtue, and in the other more security."
Lucio Anneo Séneca, Philosopher in the AD first century

"Cybercrime is the greatest threat to every profession, every industry, every company in the world."
Ginni Rometty (2015), IBM Corp.'s Chairman, President and CEO since 2012

Contents

2.1 Components of a Network Security Monitoring System 24
2.1.1 Sensors 25
2.1.2 Integrators 34
2.2 Solutions for Network Security Monitoring 41
2.2.1 IDSs 42
2.2.2 Integrators 43
2.2.3 Tool Collections 46

NSM is one of the most relevant approaches for network security. Its goal is to monitor and control the state of a given network with the aim of detecting abnormal behaviors and, when detected, to manage them in a timely manner. This is a significant challenge, since communication networks produce a huge volume of data at a high pace, following the definition of a Big Data problem [41].

Security Management is defined in [16] as "*The process of establish and maintain the security in a computer or network system*", which is composed of the following steps: *1) Prevention of security problems*, *2) Intrusion Detection*, *3) Investigation* (after one intrusion is detected), and *4) Resolution/Recuperation*. Based on this definition, and inspired in the defense cycle proposed in [19], the **NSM** cycle can be characterized as represented in Fig. 2.1: *1) Monitoring*, *2) Detection*, *3) Forensics/Diagnosis*, and *4) Response/Recovery*.

NSM requires collecting data from one or more devices and sending them to a data analysis engine [16, 19]. If a security incident is found by a detection system, it needs to be diagnosed and troubleshooted by the security team [16, 42, 218]. As soon as an abnormal activity is detected, the security

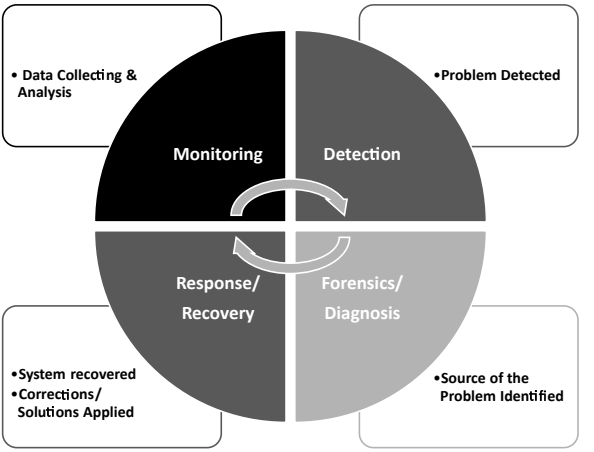


Fig. 2.1 Network Security Monitoring based in the general model of (Cyber)security Management [16, 19].

operators should be alerted. Thus, they can investigate and diagnose the detected event and take proper actions to solve it and recover the network to a normal state [16, 218].

Collecting data is not a difficult task by itself, the hard work is to make them become useful [62]. Since the monitoring gathers information from different sources/devices (usually developed for different purposes and by different manufacturers), data are commonly disparate and present different formats [62, 135]. For this reason, the captured information must be transformed as a part of the network monitoring, so that all the data can be combined and unified to the same format [135]. Another significant task is to define what kind of information should be stored. This is important not only to prevent relevant records from being lost, but also to avoid wasting storage space and processing time. In addition, security operators should be able to read and interpret the output from an NSM application, which should be in a human understandable format [19, 135].

The rest of the sections describe some relevant concepts to provide a unified vision of the topic; then, the main components for NSM are introduced. Additionally, both open source and commercial solutions for these components are reviewed.

2.1 Components of a Network Security Monitoring System

An NSM system is usually composed of one or more *sensor* and *integrator* modules. Fig. 2.2 shows the general scheme for an NSM system. A **sensor** is a mechanism that collects data from the network, generating logs or records that can be analyzed by the security team. Sensors are composed of *collectors* and, sometimes, *processors*, which allow to capture and transform the information, respectively, prior to send it to the integrator module. However, the most simple sensors might only be composed of a collector module. IDS are described later in this chapter. Sensors are configured to send the information

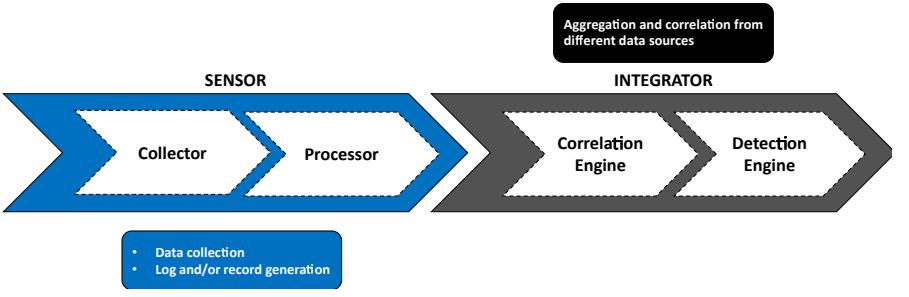


Fig. 2.2 Workflow of data through an NSM system.

to a centralized memory, where such information can be accessed. Since in typical NSM applications there is a huge amount of records to handle, it is a challenging task to detect when and where there is an attack in the network [32].

Security data can be collected from security systems or other devices, such as applications or operating system logs [135]. After data collection, these need to be structured and combined to become information (useful and understandable data). **Integrators** combine the data collected by the sensors and detect intrusions in them. First, the different records are *correlated* to extend their semantic information, yielding good models for detection of attacks or abnormal activities. This requires pre-processing the format of the data to be readable by the **correlation engine**. Afterwards, the **detection engine** detects illegitimate network traffic by means of either a model of normal operation or the signatures for known attacks.

Fig. 2.3 shows an example of an NSM system. One of the advantages of this architecture is that it is possible to build more complex systems by combining the outputs of the different modules. For example, the output of an integrator could be the input for a second integrator, making the former act as a sensor for the latter.

2.1.1 Sensors

A sensor is a device for collecting and transforming data. The data source can be either *active* or *passive* [135]. **Active data sources** are those that introduce additional network traffic in order to measure features such as connectivity or delays. Examples of these data sources are the commands `ping` or `traceroute`[1]. **Passive data sources** are those that obtain information from the network without introducing additional network traffic. Passive data sources can be grouped according to their origin as: *i) Network Traffic*, *ii) Security* and *iii) Logs and State*.

[1] See manuals in https://linux.die.net/man/8/ping and https://linux.die.net/man/8/traceroute.

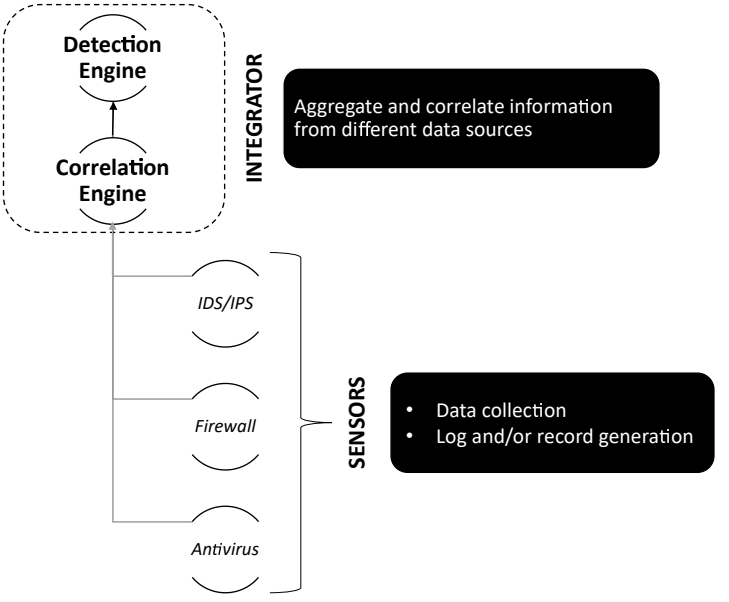


Fig. 2.3 Example of the general scheme for an NSM system.

i) Network Traffic Sensors

Data can be directly collected from the network in different formats. Some of these formats are *packets*, *sessions* (traffic flows) and statistics, which are described below.

Packets. Each communication that uses the TCP/IP protocol stack is split in packets, which are individually routed to their destination [16, 19, 134]. Thus, a packet is the basic information unit in the Internet. Packets contain detailed information of the communications, so that they are really useful for a deep investigation about security incidents (forensics) [32].

Sensors usually capture packets using a programming library such as ***libpcap*** [108] and store them for later analysis. The most common format for

the storage is `pcap`, a binary format that can be read by almost all *sniffers*[2] and traffic analysis tools [16, 19, 62, 135]. There exist several tools for collecting network traffic. The most popular are ***Wireshark*** [62, 63] and ***tcpdump*** [108], which listen in a network interface and then display the collected network traffic. They analyze the raw data from complete packets, displaying their information in an understandable format to users. **Wireshark** offers a GUI to explore packets, together with a command line tool, ***tshark***, while **tcpdump** only provides command line options [16, 19, 20, 62, 134, 135]. **tshark** provides a more powerful and complex engine than tcpdump for data analysis. However, in practice, tcpdump is the most used since it is more simple, allowing it, for example, to do the analysis on the fly during the capture of the data. On the contrary by its own developers, using Wireshark during the capture is not recommended, since there might exist some vulnerabilities in its syntax that can be used by attackers. Wireshark is more powerful for the analysis after the capture and can also be used to obtain traffic statistics [63].

The main drawback of capturing packets is that it implies a huge volume of information, rendering it impractical for long captures. An alternative solution might be filtering or limiting the size of the capture. These filters can be applied both during and after the collection [62].

Sessions (Traffic Flows). The information extracted from traffic flows provides a higher abstraction level, reducing the volume of data stored in comparison to packet captures while still allowing to have a considerable amount of information. Sessions can be collected, for example, in routers. One of the most extended protocols is `NetFlow`, which is a standard developed by Cisco Systems to extract and send information of traffic flows [60]. Although `NetFlow` was not originally developed for cybersecurity, it is widely used in this context, since it provides a highly valuable summary of the sessions and it can be easily interpretable. Two of the most relevant tools used for

[2] A **sniffer** is a program that collects and analyzes packets in a communication network.

collecting and analyzing sessions information are ***Argus*** and ***nfdump*** [62]. **Argus** collects and transforms session data [24], which are visualized and analyzed by using **Ra** client [62, 159]. **nfdump** is a set of tools (including ***nfcapd*** and the homonymous command ***nfdump***) for collecting and processing `NetFlow` data through the command line. **nfcapd** *storages* `NetFlow` data while **nfdump** *reads* the files stored by nfcapd using a syntax similar to that of tcpdump [62, 153].

Traffic Statistics. This information is related to certain features of the network, such as traffic volume or type of traffic, among others. Statistics do not allow to perform a forensic analysis *per se*, but help security operators in their investigation, complementing the data collected by other tools. One of the most extended tools used for traffic statistics is the *Simple Network Management Packet (SNMP)* [175]. **SNMP** is an application layer protocol that allows to exchange management information among devices in a network. Wireshark and tshark can also be used to obtain traffic statistics.

ii) Security Sensors

Security sensors capture information from the output of systems specifically designed for cybersecurity purposes. In what follows, some of the most used security sensors in NSM are described.

Firewall Logs. This is one of the most useful security data sources, since it provides information about each access (failed or successful, authorized or not) to the network. One of the main advantages of firewalls is that they can be found in any network. However, one of their limitations is that log systems are not always properly configured and they frequently only store logs about blocked traffic. Additionally, the location of the firewall and the blocking rules are important to determine its effectiveness as a data source.

IDS. IDSs are one of the best known security sensors. Indeed, these are a particular type of sensor, which are composed of collector, processor and detection engine. They are classified as *Host IDS (HIDS)* and *Network IDS (NIDS)* according to the origin of the collected data [9, 103, 105]. **HIDSs** are deployed in end systems (hosts) and monitor user activity and internal process behavior [16, 103, 105, 134]. **NIDSs** first collect data from the network using any of the aforementioned network traffic sensors. Then, they analyze such data to find security violations. Both HIDS and NIDS can be classified, depending on the detection approach, in: *signature-based*, if they use patterns to detect known attacks, and *anomaly-based*, if they use models of normal behavior to detect abnormal activities [16, 81, 105, 218].

Regardless the type of IDS, once the data are received and identified as (potentially) harmful, the system alerts the security operators. Managing alerts is a complex process that involves the identification of the real cause for the alert and its location, as well as to investigate related contextual information (*e.g.*the previous history of the Internet Protocol (IP) involved). In general, the main drawbacks of signature-based IDSs are the need for frequent updates of the signature database, and the inability to detect *zero-day* attacks [32, 105]. On the other hand, the main challenge for anomaly-based IDSs is to reduce the amount of false alarms, which can be performed by prioritizing and/or visualizing the events [32, 62, 135]. This can also be achieved thanks to existing lists that contain events likely reported as false positives, which allow to avoid escalating those events as alarms [16, 32, 103, 134].

Vulnerability data. These data are the result of running *vulnerability assessment* tools on the network and end systems. These tools unveil weaknesses and security holes that may enable an unauthorized access to the system. Two of the most extended tools for this purpose are ***Nmap*** [127] and ***Nessus*** [186]. **Nmap** (**Network Mapper**) is an open source program for port

scanning to evaluate the security of the operating systems[3], allowing to discover vulnerabilities and providing useful information about open ports and services. Although Nmap was originally developed for Linux, it is now multi-platform [19, 62, 127, 134]. **Nessus** is also a multi-platform program for vulnerability scanning in operating systems. Originally, Nessus was open source, but now it is private software (although there are open-source alternatives, such as ***OpenVAS*** (Open Vulnerability Assessment Scanner) [145]). The analysis usually starts with a port scanning, which can be done, for example, using Nmap. Once the open ports are discovered, Nessus sends a number of probes against such ports to unveil existing vulnerabilities. The results can be exported to different formats, such as plain text or XML [186].

Other useful resources that allow to obtain vulnerability data are the ***National Vulnerability Database (NVD)*** and ***Common Vulnerabilities and Exposures (CVE)*** databases. **NVD** is a public service provided by the National Institute of Standards and Technology (NIST) of the United States to enumerate and classify existing vulnerabilities in current software and hardware [62, 144]. CVE is another similar service provided by the MITRE[4] that also includes the NVD [140]. These databases offer the most updated information about known vulnerabilities in operating systems and applications/services, and their solution (if known). The vulnerabilities are usually discovered either using any of the aforementioned or similar tools.

FIM. File Integrity Monitoring (FIM) systems allow to detect changes in the files stored at the devices in relation to a base copy of such files. Some of the parameters that are checked by a FIM are the modification/creation date or the permissions of access and modification, as well as the checksum (hash) of the contents. One of the problems of this type of data source is the huge volume of data and the number of false positives that it tends to generate. Some of

[3] This is known in IT Security as **fingerprinting** machines.
[4] https://www.mitre.org/

the tools that implement FIM capabilities are ***OSSEC*** [154] and ***LogRhythm SIEM*** [125]. **OSSEC** is an open source HIDS [154]. **LogRhythm SIEM** is a commercial SIEM that also performs FIM [125].

Antivirus. These programs are used to detect and remove malware from computers. Antivirus software are usually signature and/or rule based, and they are designed to analyze computer files. Since antivirus generate logs of the activity, it is possible to use their outputs as a data source. However, this is typically challenging since the logging capabilities are often poorly developed in antivirus.

User Behavior Analytics (UBA). This is a method to detect internal threats, targeted attacks and financial fraud [124]. UBA is based in the definition of patterns in human behavior to create models of normal activity. The goal is to detect deviations or abnormalities that may be related to threats. To create the model, UBA needs to collect and analyze a huge amount of data about users behavior, which usually requires the application of Big Data techniques, Machine Learning (ML) and visual analytics. UBA outputs are a useful data source, since they complement the data collected by other sensors, helping to enrich the models.

Threat Intelligence. This is a method for the sharing of information about threats among organizations, which can also be useful to enhance detection engines. Threat intelligence uses knowledge related to the own organization, including context or risk indicators, but also existing reports about previous attacks, among other data [141]. The goal of threat intelligence is to foresee threats based in the previous experience, taking into account information both from the own and external organizations. Threat Intelligence tools are in charge of collecting this information and generate reports or alarms that can be integrated with other security mechanisms, such as SIEM systems. ***Threat***

connect [207] and ***Cyber Threat Alliance*** [7] are two commercial tools for threat intelligence, while ***Open Threat Intelligence*** [11] and ***Collective Intelligent Framework*** [64] are examples of open source solutions.

iii) Logs and State Sensors

Logs and state sensors gather information from applications or operating systems, among others. These sensors can be used either individually or to complement the information collected by other sensors that usually provide more detailed information. The sources include network management protocols, such as SNMP [175], which allows interchanging information among network devices and can be collected using for example ***Open SNMP***[5]; system logs, such as `syslog`, which can be captured with tools like ***syslog***[126]; or *Application Logs* obtained, for example, from ***Apache*** or ***sendmail***.

`syslog`. This is a protocol implemented in the application layer to generate logs related to the activities in a system. This protocol records occurrences, such as login events to a host or a server. This is also useful to launch alerts related to activities or errors in the operating system or the hardware, among others. Considering the type of resource that generates a record, in combination to the type of alert, it is possible to establish a scale of priorities, which is useful to help the security operators to manage such alerts.

Application Logs. Each application service, such as web surfing or the e-mail, is aimed at a different purpose and, as a result, it has its own format to record the logging information. ***Apache web server*** or ***Sendmail*** are only examples of applications that can generate logs. **Apache** is the most extended web server. It can provide data about the configuration of the websites as well as the databases, but also statistics about access to web pages. On the other

[5] https://sourceforge.net/projects/opensnmp/

hand, **Sendmail** is a Mail Transport Agent, which is in charge of routing the e-mails to their destination. For example, this is useful to investigate whether an affected host had exchanged any message with other machines before being compromised, and the nature of such messages.

Application Logs sensors allow anomaly detection, registering system accesses (both successful and failed), and prioritization in relation to the type of resource involved in an anomaly. This information can be useful for the investigation after an IT Security incident is detected [16, 135].

Additionally, we can consider a more generic type of data sources/ sensors that are not classified in any of the aforementioned groups: *meta-data* (*e.g.* tags related to the reputation of an IP).

Meta-data. This is a useful option to obtain additional information about the data collected by the rest of the tools, thus helping to contextualize such information. For example, one can investigate the owner of a given domain name by using `WHOIS` [101] or check the reputation associated to any URL or IP address using services such as ***MXtoolbox*** [143] or ***URLVoid*** [152] and ***IPVoid*** [151], which provide a number of services for this purpose, including `WHOIS` or blacklists [19, 62, 135]. For instance, if we have detected a suspicious IP address, we can find out its owner using the service `WHOIS`. We can combine the result of this service with the use of blacklists and/or statistics about attacks coming from certain locations to draw an hypothesis about the history of the suspicious IP address.

In spite of the attempts to provide unification models for the exchange of alert information, such as *Intrusion Detection Message Exchange Format (IDMEF)* [135, 170], one of the main problems in data collection is that manufacturers, when designing the devices and software, do not usually follow a standardized format for information logging. This implies the need of a

parsing[6] process to extract useful information from the data and homogenize different sources to a common format [32, 135, 170]. For instance, IP addresses can be located in different parts of the log file depending on the sensor. In such case, the parsing process is useful to identify the IP addresses on each available log and match them in order to combine different sources in a meaningful way. This process is needed to feed detection and visualization tools with structured information. However, there are some challenges related to the parsing, namely: *i) sensitivity of the parsing code to format changes in the sensors* in the sensors, usually caused by updates in their specifications or even their functionalities; *ii) scarcity of information* about the format used by each manufacturer; and *iii) lack of synchronization* in the timestamp of sensors, which can be especially challenging if they are distributed in different countries with distinct time zones and do not make use of synchronization services like *Network Time Protocol (NTP)* [32, 135].

2.1.2 Integrators

Integrators merge the data collected by the sensors, converting the format of the data to be readable by detection engines and, sometimes, correlating different records to extend their semantic information. Some of the types of integrators are [32]: *Security Event Management (SEM)* [20], *Security Information and Event Management (SIEM)* [87], and *Universal Threat Management (UTM)* [86].

i) Types of Integrators

SEM. This system is in charge of "*the collection, analysis and escalation of indications and warnings to detect and respond to intrusions*" [20]. Its aim is visualize and understand traffic data by using a single and unified tool that

[6] **Parsing** is the process of identifying and extracting individual parts that compose a log to obtain a logical and organized data structure [135].

combines different data sources. For that purpose, a SEM allows *pivoting*[7] among the different data sources. This means that, if there is an incident, the security operator will be able to navigate from one source to another to investigate it and obtain contextual information. Let us imagine that there is a `NetFlow` record that has been signaled as anomalous. By means of pivoting, it is possible for example to retrieve the reputation or the location of the source IP. Thus, pivoting reduces considerably the time needed to investigate a security incident (specially if the pivoting is graphically assisted) [20, 32]. One of the features that makes a SEM system to be such a powerful tool is that it allows the visualization and prioritization of the events, thus helping security operators to interpret and understand the alarms [87, 103].

SIEM. This system can be described following the definition of the SIEM market provided by Gartner [87, 112] as a system that "*analyzes event data in real time for early detection of targeted attacks and data breaches, and collects, stores, investigates and reports on log data for incident response, forensics and regulatory compliance*". Remember that SIEM systems are the combination of two different systems: SEM and SIM systems. The main difference in relation to the SEM is that a SIEM also performs reports and include features for regulatory compliance while the SEM does not necessary do that (indeed, this is a functionality usually provided by the SIM module). SIEM are the most popular (and expensive) type of integrator systems in the industry [32].

UTM. This is a type of "*multi-function network security product used by small or midsize business*" [86]. These devices have high level functionalities (multi-function gateway), which can be, for example, a firewall in the application layer of the TCP/IP and Open Systems Interconnection (OSI) models, Intrusion Prevention and Detection (IPS and IDS), antivirus, anti-spam and

[7] **Pivoting** refers to the ability of going from one data source to another.

anti-phishing [32, 82, 182]. The main advantages of the UTMs are their reduced cost and complexity, while the drawbacks are that UTMs usually cannot correlate events.

ii) Detection Engine

The goal of this engine is to detect suspicious behavior in the data by the integrator system (see Fig. 2.2) after the combination of the data coming from the set of sensors [32]. The volume of data to be analyzed can be reduced by filtering or grouping data, and/or by *feature extraction*[8], considering only those features that are interesting. **PCA**, for example, is a classic technique that allows to obtain new (latent) features measured by applying lineal transformations to the original data [218].

The detection engine can be defined either *Manually-based* or *using ML and Exploratory Data Analysis (EDA)*. Fig. 2.4 represents the ways of applying Data-Driven (DD) techniques, from the point of view of the security analyst, and based in the data analysis approach. The extreme case, where no DD is applied and all relies on security operators' experience, corresponds to a **manually-based** approach (left). This mechanism is applied to detect security violations using rules, which are defined manually by the security analysts. IT Security Companies have their own team of experts to analyze attacks and extract rules to identify such attacks, so that the rules can be updated in detection systems. These systems are highly reliable in situations when there is a low probability of finding a new attack, since the systems can only detect previously observed attacks. This approach is enclosed in *Signature-based* systems [81, 103].

A more autonomous mode, which in the extreme case does not need the security analyst supervision, corresponds to an automatic analysis (right in Fig. 2.4). This autonomous analysis corresponds to more traditional **ML**

[8] **Feature extraction** consists on obtaining new variables by transforming the original data records.

approaches. ML can be defined as the join of statistics and artificial intelligence to learn from data by automatically inferring and generalizing a learning model [8, 55, 68]. Indeed, this is a global term widely used to refer to the task in which one calibrates (trains) a model or algorithm to obtain a descriptive output for a given input. If the value for the output is previously known and used for the training, then the learning is named *supervised*, and it usually applies to classification and regression problems. However, if only the input data are known and the objective is to extract patterns or common behavior from the data, the learning is known as *unsupervised* [55, 179]. Mixed approaches are considered to be *semi-supervised* learning [62, 103, 157, 179, 218].

If the analyst is allowed to analyze and make any type of decision over the output of the algorithm, even modifying the detection technique, we start moving to the left, towards an exploratory detection mode in a middle point between manually-based and autonomous detection engines. The exploratory mode is a more interactive mode, which implies the security analyst supervision and still applies ML, although often this procedure is referred to as EDA. In addition, any form of DD techniques can be further classified by considering the type of machine learning they use as: *supervised* (top), *semi-supervised* (center) and *unsupervised* (bottom).

EDA consists in analyzing the data without any prior assumption [98] in order to obtain a better understanding [23, 98, 200]. "*Exploratory data analysis is detective work in the purest sense - finding and revealing the clues*" [200]. EDA combines both statistics and visualization tools (for example, scatter or bar plots [135]) to extract summarized features from the data. Data mining techniques are usually applied for a deeper examination, helping in the extraction of patterns in the data [68]. Visualization aids us to find patterns and outliers in the data, helping security operators to understand and prioritize events [32, 62, 135].

Normally, the learning process requires building a model from a training dataset by selecting some parameters, ω, that are usually optimized in con-

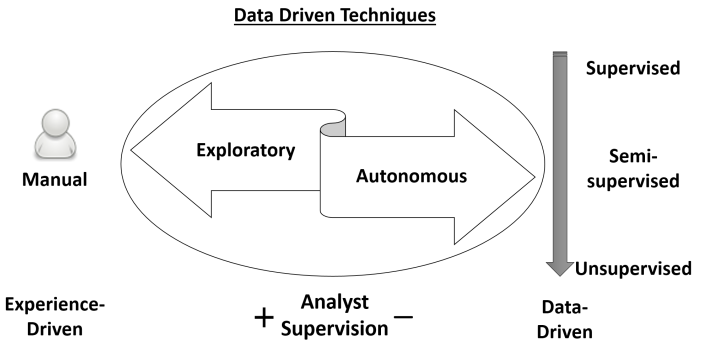


Fig. 2.4 Modes of detection engines. Manual analysis (experience-driven) is represented at the left, while the autonomous analysis (DD), is displayed to the right. Exploratory Data Analysis is shown in the middle, since it combines experience and data analysis. DD-based techniques can be vertically classified as: *supervised* (top), *semi-supervised* (center) and *unsupervised* (bottom).

secutive steps until the solution is found [8, 68]. In supervised learning, this allows the estimation of outputs for a given test labeled input. In unsupervised learning, the model provides a description of the input data [8, 55].

Supervised Learning. These methods learn features or patterns from a set of *labeled* data that are used to train a model. This model is used (in NSM) for data classification, which consists on finding the classes to what the security events belong [157, 218]. Supervised learning can be considered to be task driven, which means that, depending on the focus of the system, the way of learning can vary. For NSM, the main task is **detect known attacks**. If the system has learned a set of attacks, then it is possible to identify such attacks, which makes it a signature-based system [157, 217]. Some of the best known methods for supervised ML are [55, 157, 179, 218]:

- *Neural Networks*. Neural Networks learn or infer patterns from a set of labeled examples. Usually, to extract good classification results, a

huge amount of training data is needed. The results are very precise but hardly interpretable by humans.

- *K-nearest neighbors (KNN).* KNN algorithms compute the distance (for example euclidean distance) between the *observations*[9] to be classified and the labeled observations in a training dataset. To classify an observation, the K closest training observations are selected and, usually, the most repeated label in those is selected as the classification for the observation under study. It is simple to implement and understand, but the computational cost may be excessive if K has a high value. This method works with small datasets and a low number of features, although the variables can be reduced using some dimensionality reduction methods, such as PCA.

- *Random Forest Classification.* This method works as a collection of *decision trees*. A **decision tree** defines a set of rules to make decisions on each branch of a tree. In Random Forest Classification, each decision tree is built from random features for, finally, building a forest of low correlated trees. Random forests work under the hypothesis that a set of uncorrelated trees work better than a single tree [217].

Similarly to rule-based manual systems, supervised systems have a good performance to detect previously observed attacks. However, supervised systems are not designed to *zero-day* attacks.

Unsupervised Learning. This type of learning *does not need labels*, since the categories (*e.g.* normal vs anomaly) are obtained directly from the objects [81, 157, 218]. Some of the best known methods for supervised ML in NSM are [55, 157, 179, 218]:

[9] An **observation** is the set of properties or features that are measured for an entity. The entities of interest can be disparate (*e.g.* time intervals or devices).

- *Unsupervised Neural Networks*. These are some particular types of Neural Networks that work in an unsupervised way. A well-known example are *autoencoders*. **Autoencoders** work as noise filters and are also useful to reduce the dimensionality of the data. First, they compress the data (encode) and, then, the data are reproduced, thus removing the noise in the output (decode) [17, 100]. An example of autoencoder to implement a NIDS can be found in [139]. **Self Organized Maps** are another example of unsupervised Neural Network that also work as a method to reduce the dimensionality of the data. Self organized maps apply competitive learning [10, 160]. An example of self organized maps to implement an IDS can be found in [58].

- *Clustering*. This type of algorithm groups data according to a given similarity (shared properties) or separation criterion (dissimilar properties) [111, 178]. A well-known method is **K-means**, which groups the observations in K classes, with the optimum K being a value not known *a priori*. The procedure is an iterative repetition of a sequence of two steps: 1) compute the centroid of the K groups and 2) assign each observation to the closest group. **Unsupervised random forests** are also used as a clustering method [2].

- *(Multivariate) statistical detection*. This is another type of unsupervised learning that works in two steps: first, special causes of variation are detected and properly dealt to obtain a monitoring system that only considers the variation intrinsic to the calibration data (training model); then, new data are monitored to detect anomalies in relation to the previously trained model. *MSPC* is a well-known example of multivariate statistical detection that is applied for control in industrial processes. *MSNM* is another example of multivariate statistical detection that is applied for multivariate stastical network monitoring in network security. Both are PCA-based and are explained in Chapters 3 and 4, respectively.

Unsupervised systems are not optimized to detect a particular attack. The models are trained to distinguish between normal and abnormal traffic data, and typically applied following and EDA approach. Deciding whether the anomaly is an attack or not, as well as the type of attack or incident, is a task that should be performed by security operators and/or analysts. This task can be performed by using visualization tools that receive the output of the detection engine and transform it in useful information to help the security team deciding about the alerts. These systems are particularly practical to detect *zero-day* attacks and, in general, are more consistent with the workflow of security analysts than supervised approaches.

Semi-supervised Learning. These methods are the combination of both supervised and unsupervised learning, and make use of a partially labeled set of data [62, 103, 157, 179, 218]. Sometimes unsupervised methods are applied to discover the structure of the data. Then, supervised algorithms are applied over the data previously labeled using the unsupervised learning. Another way to apply semi-supervised learning is to use supervised learning to optimize, and then unsupervised model for the detection of a given set of attacks. Like that, the unsupervised system will be more sensitive to those attacks included during the supervised learning, being still able to detect *zero-day* attacks. This type of model optimization can be very practical, for example, a current threat has been detected in another part. If the pattern is known, it is possible to optimize the model to detect this specific type of attack while the ability of detecting *zero-day* attacks is maintained.

2.2 Solutions for Network Security Monitoring

This part of the chapter lists a collection of network security tools, both open source and commercial that include a detection engine. These tools have been grouped in IDSs, integrators, and tool collections. The latter include both

sensors and integrators as well as other security tools, yielding more complete security capabilities.

2.2.1 IDSs

As discussed before, IDSs are a type of sensor with detection engines. Indeed, they are frequently used alone, although their output may be combined with other sources in an integrator, yielding a more powerful detection system.

i) Open Source Tools

Snort. This is the most popular IDS, although it can be used also as a sniffer [103]. Snort is a signature-based NIDS, which also allows port scanning, as well as registering, alerting and providing response to any defined anomaly. `Unified2` is the output format generated by Snort, which can be generated in three modes: *packet logging*, *alert logging*, and *true unified logging* [59]. **Packet logging** is used for packet captures while **alert logging** only register IT Security events. **true unified logging** allows recording both events and packets [59].

Snorby. This is a web application for NSM, which has a GUI to manage alerts from IDSs such as Snort or Suricata. It requires the output format of the IDS to be `Unified2`, the Snort format.

Suricata. This is a detection engine for threats. Suricata is both a real-time network IDS and a network IPS. It works by network traffic monitoring and offline processing of `pcap` files. Suricata is signature-based and provides the output in standard formats, such as `YAML` or `JSON`, but it can also be configured to generate outputs in specific formats, such as `Unified2`.

Open Source HIDS SECurity (OSSEC). This is an open source HIDS that performs log analysis, integrity checking, monitoring of Windows records, and rootkit detection. In addition, OSSEC provides alerts and active responses. OSSEC is multi-platform, since it can be used in most of the Operating Systems. Although this engine has some SIEM features, such as allowing the correlation of logs from several devices and formats, and mechanisms for security policies compliance, it has been traditionally considered an IDS [154].

2.2.2 Integrators

i) Open Source Tools

Zeek (Bro). Zeek was originally developed by Vern Paxson and Robin Sommer [156] as a research work called Bro. Now, it has evolved and it is widely used by companies, as well as research and educative organizations [156]. This is a complete open source tool for NSM that permits both anomaly and signature based detection [62, 156]. Zeek collects traffic network using `libpcap`, and then the engine of events processes the data, performing a passive analysis on such data. It also allows collection and analysis of sessions of particular services. In addition, Zeek can be programmed to take actions in the evaluation of events (*e.g.* to send an email to the analyst) [156]. Although it is usually included in the IDS classification, Zeek can be considered a SEM [19, 20, 32, 156].

Prelude. This is a SIEM for Linux that collects, normalizes, combines and correlates security events. In addition, Prelude also generates reports about these events and can read any type of log [204].

Wazuh. This is a SIEM for signature-based intrusion detection, which was developed by the homonymous company [211]. Wazuh is based in OSSEC and it is used in combination with the *Elastic Stack* [71], allowing the monitor-

ing of the system for security analysis, intrusion and vulnerability detection, and providing response to security incidents, including integrity and compliance [211].

Open Source Security Information Management (OSSIM). This is a SIEM, allowing the collection, normalization and correlation of events. OSSIM was developed by Alien Vault (AT&T Cybersecurity since February 2019) [13], and it uses the Open Threat Exchange® (OTX®) [11], which allows the users to contribute and receive updated information in real-time about security information. The capabilities of OSSIM include discovering *assets*[10], assessing vulnerabilities, intrusion detection, monitoring of behavior, and correlation of events [12]. OSSIM integrates different software to provide a complete NSM solution. Among other tools, this solution is composed both of a host and a network IDS. The NIDS part provides intrusion detection and network traffic scanning. It also looks for signatures of the latest attacks, as well as malware or possible compromising of the system. The HIDS analyzes the behavior and state of the system, alerting when it suspects that there is something wrong. Similarly to other SIEMs, OSSIM allows to detect and prioritize the most important threats and anomalies [12].

ii) Commercial Tools

This part of the section covers two examples of commercial integrator systems, both included in the Gartner's "Magic Quadrant for Security Information and Event Management" for 2018 [102, 112]. This quadrant assesses the SIEMs in the market according to a set of criteria, which are mainly *Ability to Execute* and *Completeness of Vision*. **Ability to Execute** usually means that vendors are economically capable to be well positioned in the market, and **Completeness of Vision** can be seen as the ability to understand present

[10] In the case of OSSIM, asset is referred to machines.

and future needs of the market. Thus, Magic Quadrant of Gartner has four categories: *Leaders*, *Challengers*, *Visionaries* and *Niche Players*. **Leaders** have both high ability to execute and completeness of vision of the market, **Challengers** have high ability to execute but still have not found the right direction to focus on the market, **Visionaries** have good ideas and a complete vision of the market but do not have competitive ability to execute, and **Niche Players** are focused in a small segment of the market (or do not have a complete vision of it) and have a limited ability to execute [85].

Splunk. This is a commercial SIEM, which performs network monitoring and real-time data collection and correlation. Splunk also allows incident management and forensic analysis. It allows data and event analysis, providing visibility and context of the alerts. In addition, it uses Big Data techniques to integrate the data from the organization to be monitored, allowing to improve the intrusion detection by using machine learning algorithms [181]. Splunk is considered as a *Leader* in the Gartner's Magic Quadrant because it provides "*SIEM solutions that can share architecture and vendor management across SIEM and other IT use cases*" as well as "*a scalable solution with a full range of options from basic log management through advanced analytics and response*" (see Gartner's Magic Quadrant in [102]).

USM (AlienVault® Unified Security Management®). This is a commercial SIEM based in OSSIM, and it was also developed by Alien Vault (AT&T Cybersecurity since February 2019) [13]. USM is a unified platform for threat detection and policy compliance (which is one of the main differences in relation to OSSIM, see [6] for more details), as well as incident response. ***AlienVault USM Anywhere*** provides USM as a cloud service [13]. USM is considered as a *Niche Players* in the Gartner's Magic Quadrant because "*targets end-user SIEM buyers, with an emphasis on financial services and*

healthcare as well as service providers", which "*typically are mid market, not large enterprises*" (see Gartner's Magic Quadrant in [102]).

2.2.3 Tool Collections

Open Source

This type of network security tools are continuously evolving, since they are composed of a number of variate software and they are open source.

Sguil. This is a set of open source tools for network security monitoring, which allows to collect, analyze, alert and response to intrusions [19, 208]. Sguil provides a real-time interface and includes two IDSs [103, 208]. Some of the tools that compose Sguil are [208]:

- ***MySQL***, as a database service.
- ***Snort*** and ***Suricata***, for network intrusion detection and scanning as well as for logging packets and solving alerts.
- ***tcpdump***, to collect network traffic from the logs of the packets.
- ***Wireshark***, to analyze the collected packets.

Security Onion. This is a collection of open source tools, which is provided as a Linux distribution. Security Onion (SO) allows to monitor, record and manage logs, as well as to perform intrusion detection [177]. Some of the tools that compose SO are [177]:

- ***Elastic Stack***, as a search and analysis engine that also transform and centralize the data, providing visualization functionalities [70, 71].
- ***Snort***, ***Suricata*** and ***Bro***, for network intrusion detection, scanning and issuing alerts, as well as for logging of packets.

- ***Wazuh***, for host intrusion detection.
- ***Sguil***, for network security monitoring and event drive analysis.
- ***Squert***, to consult and visualize Sguil data.
- ***Cyberchef***, to encrypt, codify, compress and analyze data.
- ***NetworkMiner***, for forensic analysis.

Part I

Multivariate Statistical Monitoring

3

Multivariate Statistical Process Control

"Torture the data, and it will confess to anything."
Ronald Coase (1981), Nobel Prize in Economics in 1991

"I keep saying that the sexy job in the next 10 years will be statisticians, and I'm not kidding."
Hal Varian (2009), Chief economist at Google since 2002

Contents

3.1 Principal Component Analysis 54
3.2 Anomaly Detection 55
3.3 Diagnosis 58
3.3.1 The *Smearing* Problem 59
3.4 Batch MSPC (BMSPC) 59
3.4.1 Batch Monitoring Cycle 61
3.4.2 The Parameter Stability Problem 64

Almost fifteen years ago, Lakhina et al. proposed to apply Principal Component Analysis (PCA) for network monitoring [118]. Since then, there have been multiple works that follow this approach [26, 27, 66], including the extension to networking of a well-known anomaly detection methodology in the industry, the Multivariate Statistical Process Control (MSPC) [42]. This extension is referred to as MSNM and, due to its relevance in this work, it will be described in next chapter. This chapter is intended to review the MSPC theory.

MSPC was originally developed to monitor industrial processes. As an extension of Statistical Process Control (SPC), the aim of this methodology is to distinguish special causes of variation from common causes of variation in a process. Essentially, this means discriminating among events that are considered random variability and those that are due to an assignable cause [42, 106] The main difference between MSPC and SPC is that the former allows the consideration of multiple variables and their correlations while SPC is a univariate method and thus it only allows the monitoring of one variable at a time [42].

Using MSPC makes it possible to monitor several variables simultaneously by considering their correlation for a better model of the normal behavior and,

thus, a better anomaly detection. Due to the high dimensionality of the data in industrial processes, techniques based in latent variables, such as PCA, are usually applied in combination with MSPC.

PCA-MSPC is performed using a pair of complementary statistics that enable the indirect monitoring of a high number of variables. The statistics are computed from the PCA decomposition of the calibration data to build a model of normal operation [117, 150, 213].

MSPC is applied in two steps:

- *phase I)* The aim of this phase is to detect, diagnose and correct for special causes of variation in the process, so that only common causes of variation remain. In many cases, *e.g.* [150], this phase is limited to the removal and diagnosis of outliers, under the belief that the rest of collected data represent a stable process.

- *phase II)* After phase I, we say the process is under NOC. The second phase corresponds to the actual monitoring of new incoming data from the process. The goal is to detect excursions from the model of normal behavior in a timely manner. When an anomaly is detected, diagnosis is performed to identify its causes and classify its nature [42, 76, 150]. This allows to identify the variables related to the anomaly and helps the analysts to find the root cause of such anomaly.

The rest of the sections in this chapter describe how the PCA model is created and the computation of the statistics is performed, their use for the monitoring, and the way of performing the diagnosis for MSPC. Finally, a last section describes the BMSPC, which is intended to apply MSPC for batch process monitoring.

3.1 Principal Component Analysis

Given an $N \times M$ data matrix, where N is the number of observations and M the number of variables, PCA identifies the sub-space with maximum variance in the M-dimensional space of variables, represented by the first A Principal Components (PCs). The original variables are linearly transformed into the **PCs**. To do this, first the data matrix $\mathbf{X}$ is normalized, typically by mean-centering:

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1} \cdot \mu \tag{3.1}$$

where $\mathbf{X}_c$ represents the mean-centered data, $\mathbf{1}$ is a column vector and μ is the array containing the sample means of the columns of $\mathbf{X}$.

The data matrix sometimes is also normalized by Auto-Scaling (AS) as follows:

$$\mathbf{X}_{sc} = \mathbf{X}_c \oslash \mathbf{1} \cdot \sigma \tag{3.2}$$

where $\mathbf{X}_{sc}$ represents the auto-scaled data, $\oslash$ is the Hadamard (element-wise) division and σ is the array containing the sample standard deviations of the columns of $\mathbf{X}$.

The PCA model can be expressed as follows [117, 213]:

$$\mathbf{X_{sc}} = \mathbf{T}_A \cdot \mathbf{P}'_A + \mathbf{E} \tag{3.3}$$

with $\mathbf{T}_A$ the *score* matrix of size $N \times A$, $\mathbf{P}_A$ the *loading* matrix of size $M \times A$, and $\mathbf{E}$, the *residual* matrix of size $N \times M$. The columns in $\mathbf{P}_A$ correspond to the eigenvectors of $\mathbf{XX} = \frac{1}{N-1} \cdot \mathbf{X_{sc}}' \cdot \mathbf{X_{sc}}$.

Once the PCA model is built, the scores for a new observation, $\mathbf{t}_{new}$, are computed by projecting that observation, $\mathbf{x}_{new}$, onto the model subspace:

$$\mathbf{t}_{new} = \mathbf{x}_{new} \cdot \mathbf{P}_A \tag{3.4}$$

and the residuals, $\mathbf{e}_{new}$, are calculated using these scores:

$$\mathbf{e}_{new} = \mathbf{x}_{new} - \mathbf{t}_{new} \cdot \mathbf{P}'_A \tag{3.5}$$

3.2 Anomaly Detection

Both scores and residuals are monitored in the MSPC system using two statistics, namely, the D-statistic (D), and the Q-statistic (Q). The D-statistic is computed to monitor the model subspace [117, 212, 213].

$$D = \sum_{a=1}^{A} \left(\frac{t_a}{s_a}\right)^2 = \sum_{a=1}^{A} \frac{(t_a)^2}{\lambda_a} \tag{3.6}$$

where t_a and s_a^2 are, respectively, the *score* for the a^{th} component and the sample variance of this score. The variances of the PCs are the eigenvalues, λ_a, of $\Lambda = \frac{1}{N-1} \cdot \mathbf{T}'_A \cdot \mathbf{T}_A$.

To monitor the residuals, the Q-statistic is calculated as:

$$Q = \sum_{m=1}^{M} (e_m)^2 \tag{3.7}$$

where e_m is the residual value of the observation corresponding to the m-th variable.

If any of the statistics corresponding to a new observation is greater than a threshold, the so-called *Upper Control Limit (UCL)*, this observation is identified as anomalous. The derivation of UCL values for both statistics follows. The scores are linear combinations of the original variables; thus, according to the Central Limit Theorem, they follow a Normal distribution [150]. As a consequence, the D-statistic times a constant follows a β – distribution in *phase I* [198]:

$$D \sim \frac{(N-1)^2}{N} B_{A/2,(N-A-1)/2} \tag{3.8}$$

therefore, the corresponding UCL for the D-statistic at significance level α is given by:

$$UCL(D)_\alpha = \frac{(N-1)^2}{N} B_{(A/2,(N-A-1)/2),\alpha} \tag{3.9}$$

For new incoming data in *phase II*, the D-statistic times a constant follows an F – distribution [198]:

$$D \sim \frac{A \cdot (N^2-1)}{N \cdot (N-A)} F_{A,(N-A)} \tag{3.10}$$

and the corresponding UCL at significance level α is given by:

$$UCL(D)_\alpha = \frac{A \cdot (N^2-1)}{N \cdot (N-A)} F_{(A,(N-A)),\alpha} \tag{3.11}$$

Several procedures can be used to determine the UCL for the Q-statistic. Again, the residuals can be assumed to follow a multivariate normal distribution. Jackson and Mudholkar showed in [107] that an approximate critical value at significance level α is given by:

$$UCL(Q)_\alpha = \theta_1 \cdot \left[\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0-1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{3.12}$$

where $\theta_n = \sum_{a=A+1}^{rank(\mathbf{X})} (\lambda_a)^n$, with $rank(\mathbf{X})$ the rank of $\mathbf{X}$ and λ_a the eigenvalues of matrix $\frac{1}{N-1} \cdot \mathbf{E}' \cdot \mathbf{E}$, $h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}$; and z_α is the $100 \cdot (1-\alpha)\%$ standardized normal percentile.

Alternatively, the approximation based on the weighted chi-squared distribution proposed by Box can be used [22]. Control limits for the Q-statistic that distinguish *phase I* and *phase II* can also be found in [76].

To achieve adequate performance in *phase II*, it is highly recommended to re-adjust the control limits using the calibration data on a leave-one-out cross-validation basis [44, 161]. The limits are raised or lowered so that the

Overall Type I (OTI) risk equals the imposed significance level α [44, 95]. Following the definition in [150], the OTI is the percentage of false alarms in the Normal Operation Condition (NOC) calibration observations:

$$OTI = 100 \cdot \frac{\#f}{N}\% \tag{3.13}$$

where $\#f$ is the number of single observations where either of the statistics computed surpasses its control limit in the NOC calibration data.

The **monitoring charts** are plots used to represent the statistics and the control limits. They help us to visualize and interpret the value of the statistics. For a better understanding, let us take as an example the monitoring chart displayed in Fig. 3.1.

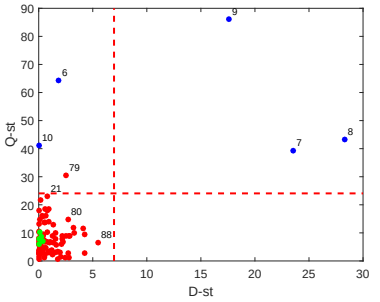


Fig. 3.1 D- and Q-statistics and their corresponding theoretical control limits.

To build the example, first we generate a 100×10 calibration matrix at random. Then, we also created a 10×10 test matrix following the same correlation pattern as in the calibration. The value of the first two variables in the last five test observations is increased to make them anomalous. Fig. 3.1 shows the D- and Q-statistic, as well as the corresponding theoretical UCLs. The UCL for the D-statistic is represented as a vertical dashed line, while for the Q-statistic this is represented as an horizontal dashed line. Those observations that are to the right of the UCL for the D-statistic are abnormal,

although still follow the correlation structure in the PCA model. Those observations that are above the UCL for the Q-statistic are abnormal, and break the correlation structure in PCA.

Fig. 3.1 shows the statistics for the calibration data colored in red. All the calibration data except observation #79 are under the UCL for the Q-statistic and left to the UCL for the D-statistic. Observation #79 can be considered as a mild outlier, since the Q-statistic is above its UCL. For the test, we can observe the observations under NOC represented in green, while those that are anomalous are represented in blue color. Observations #6 and #10 show anomalous values in the Q-statistic, and observations #7 to #9 show anomalous values both in the D- and the Q- statistic.

3.3 Diagnosis

Once an anomalous behavior is detected in a process, it should be diagnosed. This is usually done by analyzing the contributions of the monitored variables to the value of the statistics exceeding the control limits. Those variables with a higher magnitude in their contribution are considered to be related to the anomaly. The most used diagnosis methods are the Contribution Plots (CP) and the Reconstruction-Based Contributions (RBC) [4, 5, 69, 117, 150, 212, 213]. Other alternatives, such as observation-based Missing-data method for Exploratory Data Analysis (oMEDA) [29], have also been proposed. These multivariate diagnosis methods are reviewed in Chapter 7.

We use the previous example to illustrate how diagnosis methods work. We apply oMEDA to diagnose the anomalous observations in recall Fig. 3.1. As a result, a bar plot is obtained in Fig. 3.2. This plot represents the contribution of each variable to the diagnosed anomaly. We can observe how the method highlights clearly the two first variables, which are those what we made to be anomalous.

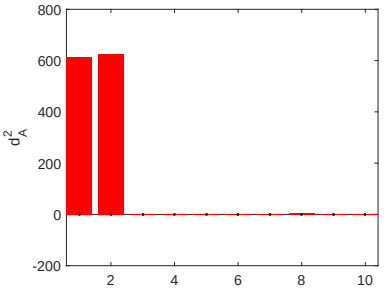


Fig. 3.2 Example of diagnosis applying oMEDA, obtained using the ***MEDA-Toolbox*** [43].

3.3.1 The *Smearing* Problem

The state-of-the-art diagnosis methods are multivariate and they can suffer from the *smearing* effect: misdiagnosis owing to the spread of the contribution from the variables affected by an anomaly to those not affected by it [114, 212].

The *smearing* problem results in a more complex diagnosis process. A wrong diagnosis may lead to an incorrect identification of the source of the anomalies, affecting also the triage and prioritization of such anomalies, which is of main interest for this work. The diagnosis and the smearing problem are treated in detail in Chapter 7.

3.4 Batch MSPC (BMSPC)

Batch processes are a special type of industrial processes that are important for many production areas, such as the chemical, pharmacy and food industries [148–150]. A **batch** refers to a process composed of a number of phases and steps that are repeated cyclically, following a specific a recipe [52]. Each of these repetitions is called a batch. The duration of the batch production can vary from batch to batch, depending on the initial conditions, as well as other factors [52, 91, 92]. The process variables can be monitored

either *online* (during the batch processing) or *offline* (after the batch processing). The main disadvantage for the latter is that monitoring results might not be available before starting the processing of next batch and, thus, the reaction to a process failure is delayed [52]. Batch processes allow a tight control of the process, which in the end is translated into a greater economical impact [52, 91, 115, 148–150].

Batch processes present characteristic cycles that are repeated during the execution of each batch. In this work, we use the term **cyclo-stationarity** to refer to this structure. This makes it necessary to adapt the existing procedures in MSPC to batch monitoring (BMSPC) [52, 115, 148–150]. The network traffic data are also characterized by the presence of cycles in their activity. For example, one can identify similar patterns during the day or the night, but also among working days or weekends. These patterns are periodically repeated for the same network. Again, we can observe the analogy between industrial processes and network traffic data. For this reason, we also pay special attention to BMSPC in this work.

Typically, a batch process is monitored by collecting measurements on J variables through the batch production. After collecting measurements on I batches, a data set of $I \text{x} J \text{x} K_i$ measurements is available. The cyclo-stationarity of the network data can be considered by re-arranging the observations in a three-way matrix[1]. To build a model from batch data, a number of modeling steps are typically performed: *1) Alignment*, *2) Pre-processing*, *3) Unfolding*, *4) Calibration*, and *5) Detection* [92]. The latter is performed following the MSPC theory: to detect special causes of variance in *phase I* and anomalies in *phase II*, once the NOC model has been obtained. If an anomaly is detected, like in MSPC, the Diagnosis is also performed.

[1] In network data both a day or a week can be considered a cycle. In this thesis we consider each day the cycle of interest.

3.4.1 Batch Monitoring Cycle

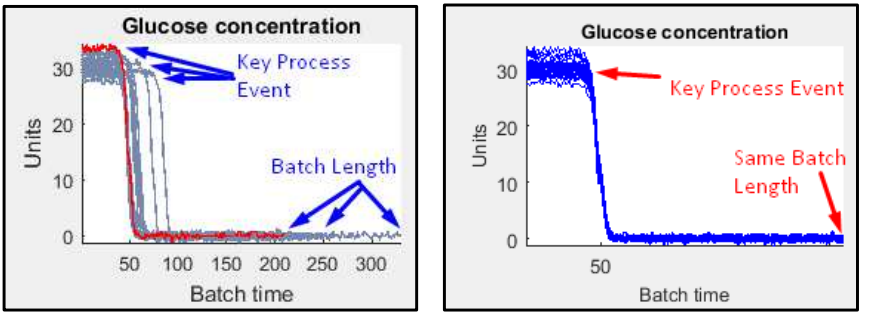
1) Alignment Step

For batch process monitoring, all the process variables should be aligned to the same pace and all the *key process events*[2] should also be synchronized (see Fig. 3.3). An incorrect synchronization has been proven to affect the correlation of the data [91, 93, 95] as well as the stability of the model parameters [92]. The propagation of these problems causes the anomaly detection model to be degraded. Additionally, it it is taken into account that the batches might not have the same length, it is evident the need of applying methods for alignment and synchronization of batches [91, 93, 95]. With such purpose, several approaches can be found in the literature. According to [91, 92], they can be classified in three groups, considering if they are based on: *i)* compressing and/or expanding intervals in the variable trajectories using linear interpolation, *ii)* feature extraction, or *iii)* expanding, compressing and translating intervals in the variable trajectories [92]. After the application of such methods a three-way structure of dimensions IxJxK is obtained [91]. As an illustrative example, Fig. 3.3 (a) shows the first variable of the *Saccharomyces cerevisiae* process simulation without any alignment, while Fig. 3.3 (b) represents the effect of applying an alignment method (the multi-synchro algorithm [93]) in the same variable. We can observe that the key process event is synchronized and the batch length is the same after the alignment [89, 91].

2) Pre-processing Step

Synchronized data are usually pre-processed for several reasons, namely to: *i)* remove offsets, *ii)* account for differences of magnitude across variables, and *iii)* give certain weights to variables depending on the nature of data.

[2] A **key process event** defines the moment in which each step of a process takes place (when the step starts and ends). Key process events usually vary from batch to batch (see Fig. 3.3 (a)).



(a) (b)

Fig. 3.3 Visual representation of the first variable of the *Saccharomyces Cerevisiae* process simulated using ***MVBatch*** [89] toolbox (a) without applying any synchronization or alignment, and (b) after applying the multi-synchro algorithm [89, 91].

In batch process monitoring, there are two generally accepted pre-processing methods: *Trajectory Centering and Scaling (TCS)* [149] and *Variable Centering and Scaling (VCS)* [214][3], which are detailed in Chapter 6.

3) Unfolding Step

Once the data are aligned, synchronized and pre-processed, they need to be transformed from three-way into two-way to apply bilinear techniques like PCA. This is termed the unfolding of the data. There exist different approaches to unfold the data. A thorough study on the differences among these methods in terms of the process dynamics captured by the bilinear model can be found in [46]. For implications on quality prediction, the reader is referred to [47].

The main approaches to unfold the data into a single two-way matrix are: *i) batch-wise* [148], *ii) variable-wise* [214], and *iii) batch-dynamic* [53]

[3]Note that TCS and VCS are terms that were introduced in [92] to refer to pre-processing techniques associated with the batch modeling approaches originally presented by Nomikos et al. [149] and Wold et al. [214], respectively.

unfolding (see Fig. 3.4). **Batch-wise** unfolding is performed in the batches direction, so that the result is an $I \times KJ$ matrix. **Variable-wise** unfolding is carried out in the variables direction, so that the result is an $IK \times J$ matrix. **Batch-dynamic** unfolding where a number of *lags*[4] is defined can be seen as an intermediate solution to batch- and variable-wise unfolding. The higher the number of lags, the more similar to a batch-wise approach[5]. On the contrary, the lower is the number of lags, the more similar is batch-dynamic to a variable-wise unfolding[6] [46, 52]. Other approaches unfold the data in several two-way matrices, for example by creating K local models [163]. This method can also be combined with other approaches [121, 162]. Finally, other alternative is the multi-phase approach, which consists in calibrating independent models for different stages or phases in the process [48, 201].

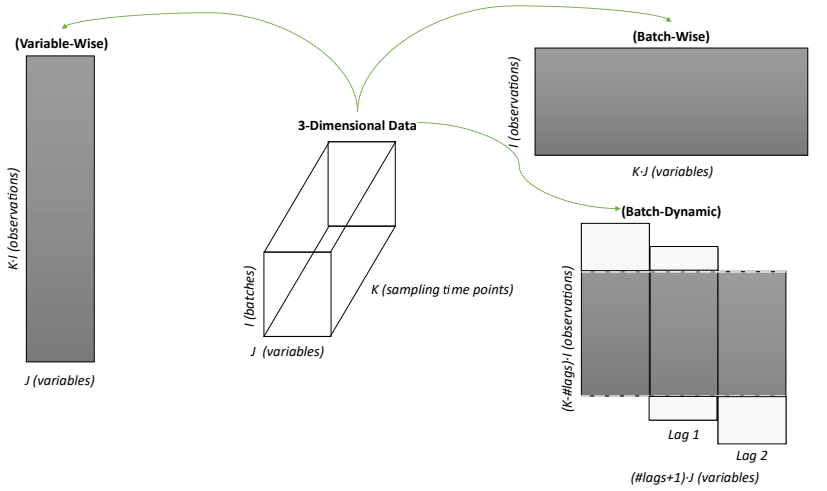


Fig. 3.4 Main types of unfolding into a single matrix. Figure adapted from [46, 52].

[4]Each **lag** defines the value of the current sampling time as the previous one.

[5]The maximum number of lags, (#*lags*), is $K-1$. In this case, the batch-dynamic approach corresponds to a batch-wise unfolding.

[6]If there are no lags, which is an extreme case, the batch-dynamic corresponds to a variable-wise unfolding.

4) Calibration Step

Unfolded data are typically modeled by projection techniques to latent structures, such as PCA or Partial Least Squares (PLS) regression. Depending on the objective, the choice of the multivariate statistical model will differ. The interest in this thesis is limited to PCA.

5) Detection Step

Similar to the MSPC, in BMSPC a monitoring scheme is designed based on the estimation of the D and Q statistics [158]. The control limits for their control charts are computed from NOC process data, and can be re-adjusted for a given imposed significance level by cross-validation [49]. In BMSPC, *phase I* and *phase II* are also applied [75].

During *phase I* the goal is to distinguish between the common and special sources of variability across batches. Those variations that are considered to be assignable in the process need to be corrected and eliminated, so that finally we can build a model from a NOC process [89]. After the calibration model has been created, in *phase II*, incoming batches are monitored to determine whether the process remains under NOC or not.

6) Diagnosis Step

Similar to MSPC, after detecting an anomaly in a process, either in *phase I* or *phase II*, those observations that make the statistics exceed their corresponding UCL need to be diagnosed. In BMSPC it is possible to apply the same diagnosis methods as in MSPC (see Section 7.1).

3.4.2 The Parameter Stability Problem

Each of the steps in the data pipeline until the calibration model is created has influence on the stability of the monitoring system.

Previous studies show that the lack of stability affects the quality of the PCA model and the detection performance of MSPC [92]. More precisely, the effects of the synchronization method and the unfolding approach in the *parameter stability* have already been evaluated. The **Parameter stability** is inversely related to the uncertainty: the greater the uncertainty in a model, the lower the parameter stability and, thus, the less reliable the system is. It is desirable to reduce the parameter uncertainty in the model.

One of the critical factors on the parameter stability is the *Ratio number-of-Observations-to-the-number-of-Parameters (ROP)* [92]: a high ROP is pursued to achieve a low parameter uncertainty. A thorough study on the implications of each of the modeling steps on the parameter stability can be found in [92]. In particular, the cited work shows that a main source of model instability is the over-parameterization in the TCS pre-processing. The parameter stability is treated in detail in Chapter 6.

4

Multivariate Statistical Network Monitoring

"I have no special talent. I am only passionately curious."
Albert Einstein, Nobel Prize in Physics in 1921

"Intelligence is the ability to adapt to change."
Stephen Hawking, Physicist

Contents

4.1 MSNM 69
4.1.1 Parsing 70
4.1.2 Fusion 71
4.1.3 Detection 72
4.1.4 Diagnosis 74
4.2 Extensions on the MSNM application 74
4.2.1 Extensions in the Fusion Step 76
4.2.2 Extensions in the Detection Step 78
4.2.3 Extensions in the Diagnosis Step 82
4.2.4 Extensions for Big Data 82

The use of PCA to build anomaly detection engines in the context of NSM was proposed by Lakhina *et. al* in a pioneering work in the early 2000's [118]. The main goal of the proposal is to distinguish normal from abnormal network traffic by means of using a PCA model and to diagnose the root causes of anomalies. Lakhina applied PCA over **counts** of packets and bytes in the traffic of the network. A later work from the same authors [119] proposed to use the entropy as a measure of change in the counts. As already discussed, PCA creates two sub-spaces: one corresponding to the model and other to the residuals. A strong assumption of [118] and [119] is that the model sub-space corresponds to normal variability, while residuals correspond to abnormal variability. Besides, the data under monitoring was the same used to create the model. This approach has as main drawback that anomalies may pollute the PCA model, degrading the quality of the anomaly detection systems [42]. There have been several modifications of the original PCA-based approach with the aim of solving some of its weaknesses [26, 27, 66,

168]. In the last years, other research works on multivariate analysis for security-related anomaly detection have opted for combining PCA with other detection schemes. For example, Aiello et *al.* [3] combine PCA with mutual information for profiling DNS tunneling attacks; and Fernandes et *al.* [74] combine PCA with a modified version of dynamic time warping for network anomaly detection. These authors also propose an alternative approach based on ant colony optimization. In [109], Jiang et *al.* apply PCA over a wavelet transform of the network traffic for network-wide anomaly detection. A similar approach with multi-scale PCA is used in [54]. Xia et *al.* [216] propose an algorithm based in the Singular Value Decomposition (SVD) which is combined with other techniques for anomaly detection by considering the cyclo-stationarity of the data.

Most of these proposals still suffer part of the disadvantages of the original work [118]. This motivated the development of the *Multivariate Statistical Network Monitoring (MSNM)* methodology, which was introduced in 2015, based in the MSPC theory (see Chapter 3) [42]. The rest of the sections of this chapter describe the MSNM methodology and related extensions.

4.1 MSNM

MSNM is an extension of the MSPC theory for NSM. It can be enclosed in the unsupervised learning paradigm, allows to combine traffic data with other security data sources [30] and it has shown a detection performance comparable to the state-of-the-art machine learning methodologies [38]. The most relevant advantage of MSNM in relation to such methodologies is its ability to provide diagnosis support [36, 38].

MSNM has four steps: *1) Parsing*, *2) Fusion 3) Detection*, and *4) Diagnosis* [42], which are summarized in this section.

4.1.1 Parsing

A main difference of the data used in MSPC and network monitoring data is that network data usually come in the form of logs of different sensors structured in different formats, as discussed in Chapter 2. For this reason, these data need to be processed and transformed into a common format. After this processing, a set of features is extracted, which is used by the detection engine to build a model for later anomaly detection. The process of identifying which features we need to extract is known as **feature engineering**. Features can be obtained either *directly* or *indirectly* from logs. Thus, **direct** parsing refers to those features that do not need any transformation prior to the extraction (*e.g.* number of packets per connection, as it can be seen in the 5th column of Fig 4.1). **Indirect** parsing refers to those features that are not in the log but can be obtained as a combination and/or transformation of some information in them (*e.g.* number of connections per minute), which requires the combination of several rows in Fig 4.1.

Date first seen	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Packets	Bytes	Flows
2016-04-25 00:02:01.880	1.460	TCP	211.42.204.231:59042	->	42.219.156.211:80	6	718	1
2016-04-25 00:02:01.880	1.436	TCP	42.219.156.211:80	->	211.42.204.231:59041	4	985	1
2016-04-25 00:02:01.880	1.436	TCP	42.219.156.211:80	->	211.42.204.231:59042	4	934	1
2016-04-25 00:02:01.880	4.484	TCP	211.42.204.231:59043	->	42.219.156.211:80	5	658	1
2016-04-25 00:02:01.880	4.460	TCP	42.219.156.211:80	->	211.42.204.231:59043	4	934	1
2016-04-25 00:02:01.880	4.484	TCP	211.42.204.231:59045	->	42.219.156.211:80	5	656	1
2016-04-25 00:02:01.880	4.460	TCP	42.219.156.211:80	->	211.42.204.231:59045	4	940	1
2016-04-25 00:02:01.880	1.032	TCP	42.219.159.95:64578	->	202.168.135.171:445	3	144	1
2016-04-25 00:02:01.880	4.416	TCP	211.42.204.231:59044	->	42.219.156.211:80	5	652	1

Fig. 4.1 Example of raw data source for MSNM: `NetFlow` record from the UGR'16 [130].

In MSNM [30], the use of counters in the work of Lakhina *et al.* [118] was generalized to consider the disparity of data sources. This is termed the *feature-as-a-counter* approach: the combination of data counters with multivariate analysis. Each feature contains the number of occurrences for a given event during a period of time. The combination of different sources of information is simplified thanks to this feature definition [41].

Fig. 4.2 shows an example of the counters corresponding to the example depicted in Fig. 4.1. These counters indicate the number of occurrences of

the features for each observation, defined as a time interval of one minute. Like that, for instance, we can observe that most of the source and destination IPs in the capture are public, since the number of counters is greater than those for the private IPs (columns #1 and #4 correspond to private source and destination IPs, respectively; while columns #2 and #5 correspond to public source and destination IPs).

srcip_private	srcip_public	srcip_default	dstip_private	dstip_public
7	43048	0	0	43055
13	41436	0	0	41449
9	41766	0	0	41775
11	43002	0	0	43013
7	41978	0	0	41985
12	45965	0	0	45977
9	41230	0	0	41239
11	40891	0	0	40902
12	40692	0	0	40704
9	40521	0	0	40530

Fig. 4.2 Counters corresponding to the example in Fig. 4.1.

4.1.2 Fusion

Due to the fact that each data source generates its own set of features or variables during the parsing step, these need to be combined to obtain a single stream of featured data. Since each source can have different sampling periods, the features may need to be stretched/compressed to a common sampling rate, so that they can be finally merged. Once the sampling rate is homogenized, the features are organized in a single high dimensional data matrix.

Fig. 4.3 represents the fusion step conceptually: first, the features are extracted from each data source. The result is a number of data matrices with the features and observations corresponding to each data source (represented in different colors). It can be observed that the dimensions of these matrices is

not necessary the same. Second, after the fusion step, all the matrices are concatenated into a single matrix with an homogeneous number of observations and as many features as the sum of the features of the original data sources.

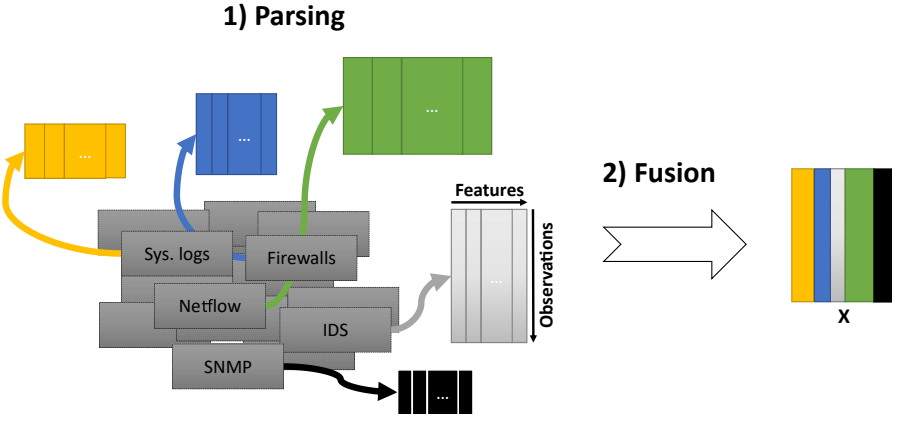


Fig. 4.3 Conceptual representation of the fusion step.

4.1.3 Detection

Once the *parsing* and the *fusion* steps are finished, the MSPC steps can be followed. PCA is applied with the objective of obtaining the subspace of maximum variance in the M-dimensional variable space. This allows the detection of anomalies by the computation of the D-statistic and the Q-statistic. The use of PCA, in combination to these statistics, reduces the complexity of network security monitoring.

In addition, for each observation n, the D-statistic and the Q-statistic can be combined into a single score, the *Tscore*. This measure allows to take into account at the same time both the D- and the Q- statistic. In NSM, the interest is more on anomalies triage rather than detection. Thus, the Tscore helps security operators to prioritize alarms, since they only have to take into account a single statistic. Fig. 4.4 shows an example of detection using the

Tscore. In this example, we can observe that five observations are clearly different to the rest. Observations are prioritized according to their Tscore to perform the diagnosis and investigate the root causes of the corresponding anomalies.

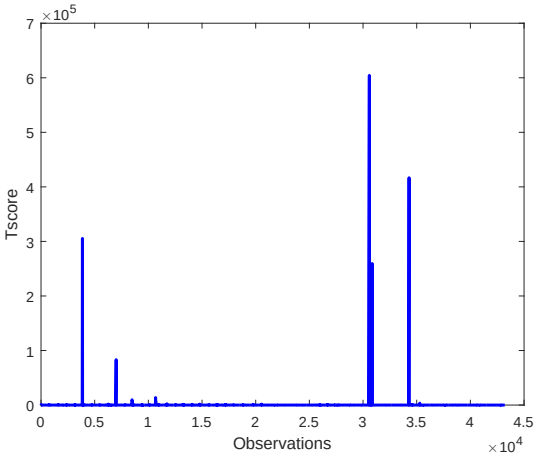


Fig. 4.4 Example of detection using the Tscore.

The Tscore is defined according to the following equation [36]:

$$T_n = \alpha \cdot \frac{D_n}{UCL^D_{.99}} + (1-\alpha) \cdot \frac{Q_n}{UCL^Q_{.99}} \tag{4.1}$$

where $UCL^D_{.99}$ and $UCL^Q_{.99}$ are the upper control limits for the D-statistic and the Q-statistic computed from the calibration data [42], which are computed as 99% percentiles, and $\alpha \in [0,1]$ is a weighting factor for the combination of both statistics [36].

Like MSPC, MSNM is applied in two phases: *phase I*, which is intended to identify and correct for special sources of variability in the traffic data; and *phase II*, where the new data are monitored to detect and diagnose anomalies based on the model of normal operation. Since phase I and phase II are focused

in different objectives, the weighting factor α in Equation (4.1) is defined in different ways for each of them. In phase I, α corresponds to the percentage of captured variance by the PCA model, since assignable sources of variance affect the model. In phase II, α can be defined as the ratio between the number of selected PCs, A, and the number of variables, M, since anomalies can be found in any direction of the space of variables.

4.1.4 Diagnosis

Once an anomaly is detected, a diagnosis is performed to identify the features related to the anomaly. This is important to help in the identification of the root causes of the anomaly. Let us consider as an example Fig. 4.5, which shows the oMEDA diagnosis for the *Scan44* attack in the UGR'16 dataset [130]. We can observe a bar plot where the most highlighted features are expected to be related to the anomaly[1]. Thus, the diagnosis is a useful tool to prioritize the investigation of the anomaly, enabling the security team to response in a faster way.

4.2 Extensions on the MSNM application

Since the first proposal of MSNM, there has been a number of extensions [33, 36, 40, 129, 192, 195] to improve the performance of the methodology and enhance it with new functionalities. Most of these extensions are related to single step of the MSNM. Fig. 4.6 depicts a diagram of the extensions. They are also described later in this section. This work has contributed to most of the extensions, leading those related to pre-processing (the PARAMO approach) [192] and diagnosis [195] (a methodology for the comparison of

[1] *dport_citrix* and *dport_msnmessenger* are those with the highest magnitude, which are easily visible in the plot. Then, features *dport_register*, *dport_kpasswd* and *sport_citrix* have also a high magnitude. This diagnosis points out a port scanning attack.

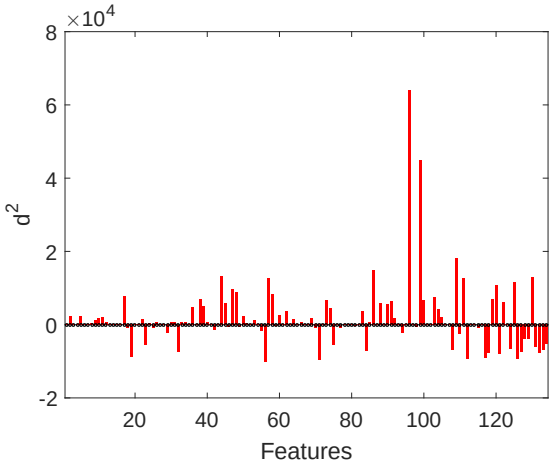


Fig. 4.5 Example of diagnosis using oMEDA.

diagnosis methods, and the U-Squared method), which are detailed as main contribution of this thesis in Chapters 6 and 7 respectively.

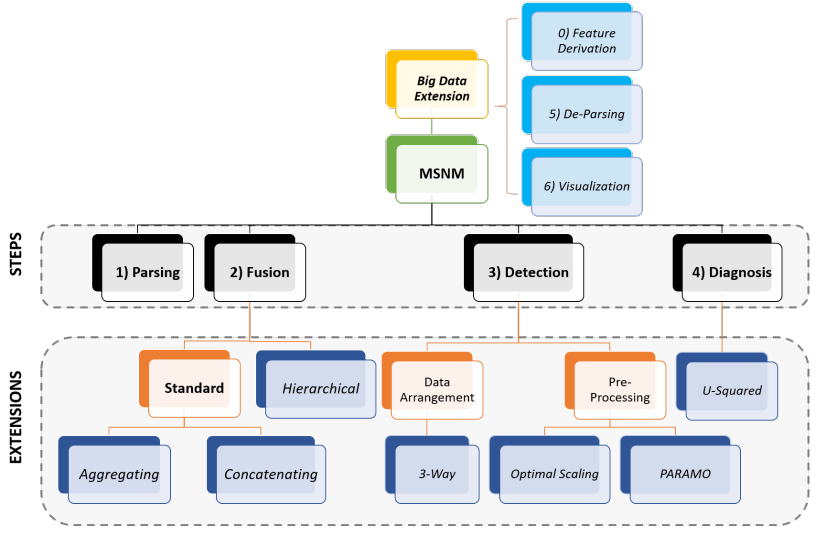


Fig. 4.6 MSNM steps and the main extensions.

4.2.1 Extensions in the Fusion Step

There are different levels of data fusion: *low*, *middle* and *high* [61, 67]. In the **low-level**, the fusion is performed in the data, prior to the model building. In MSNM, we understand as low-level fusion when it is done at counts level. In the **middle-level**, the fusion is carried out after feature extraction from the data, after applying PCA. Finally, in the **high-level**, the fusion is performed on the output of several anomaly detection, regression or classification systems. An example can be to fuse the statistics obtained after applying MSPC [61, 67, 180].

Standard Fusion

When one needs to monitor a communications network, it is usual that some type of data are collected from many sensors (see Chapter 2) in a distributed fashion. After the parsing and fusion steps, performed locally in each sensor, these data can still be maintained in individual but homogenized matrices (with the same number of features, M, and observations, N). Let us call the resulting matrices $\mathbf{X}^i$, where $1 \leq i \leq S$, and S corresponds to the number of monitored sensors.

In standard MSNM, these data are generally combined following a low-level fusion in a single matrix, by concatenating the features in a single matrix, $\mathbf{X} = [\mathbf{X}^1\ \mathbf{X}^2\ ...\ \mathbf{X}^S]$ with dimensions $NxSM$ (recall Fig. 4.3). In what follows we will call this approach **C**-fusion, term that comes from **c**oncatenating. If the sources are of the same type (*e.g.* two routers), another alternative is to fuse by **aggregating** (summing) the counters corresponding to each feature. The result is a matrix $\mathbf{X} = \sum \mathbf{X}^i$ with the same dimensions as the individual matrices, NxM. In the following we will call this approach **A**-fusion, term that comes from **a**ggregating. Finally, the PCA model is built in both cases as usual from the matrix $\mathbf{X}$.

The **C**-fusion allows to identify the source of an anomaly using diagnosis tools, that is, the sensor collecting the anomalous data, while this is not possible using the **A**-fusion. Furthermore, different to the **A**-fusion, the **C**-fusion allows the consideration of different types of data sources (which is the most realistic case in NSM). As a downside, the concatenation of the features increases resources required for the computation of the model, while the aggregation simplifies this calculation. Also, concatenation increases the number of variables, and thus the model uncertainty. The performance in terms of anomaly detection is evaluated for both approaches using real network data in Chapter 8.

Hierarchical MSNM

An alternative approach for data fusion in MSNM was proposed in [129]. This is a hierarchical model (**H**-fusion), which consists on the computation of the statistics in different layers of the hierarchy[2]. Both statistics and data can be combined in different parts of the hierarchy using one or more integrators. Examples of this approach are shown in Fig. 4.7. Fig. 4.7 (a) shows a hierarchy with a single integrator. In the low layer, each sensor is modeled individually using MSNM. The statistics of the different sensors are integrated at the top layer, where another MSNM model is computed. If an anomaly is detected at the top layer, the diagnosis is used to identify the original source of the problem. Then, in the low layer, the diagnosis is repeated to identify those variables related to the anomaly. Fig. 4.7 (b) shows a more complex hierarchy with two integrators and three layers. Regardless the complexity of the hierarchy, the mechanism for detection/diagnosis remains as in the first example.

In general, the hierarchical union of the data shows the following benefits:

[2] In hierarchical MSNM, level refers to the position, layer or stage in the hierarchy rather than the aforementioned level of fusion.

- Preserving ***C**-fusion advantages*: prioritization and identification of the location and/or source of the anomaly.
- *Volume and time consumption reduction* of the data needed for the monitoring (see Appendix (D)).
- *Scalability*, since a number sources can be added to the architecture of the hierarchy, yielding more possible scenarios.
- *Privacy enhancing*, since it is possible to apply high-level fusion in the top layers of the hierarchy, avoiding to send features to the integrator, which might not belong to the same organization that is generating such features.

The hierarchical MSNM is evaluated using real network data and compared with the two standard fusion alternatives in Chapter 8.

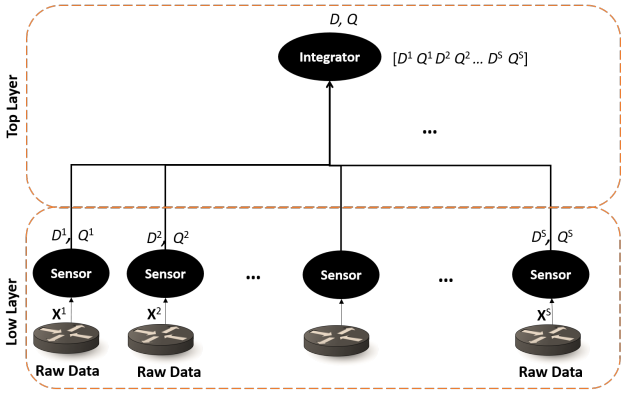
4.2.2 Extensions in the Detection Step

The enhancements presented in this section correspond to *i)* the three-dimensional arrangement of the data taking into account the cyclo-stationarity; and *ii)* the pre-processing of the data (*Optimal Scaling* and PARAMO).

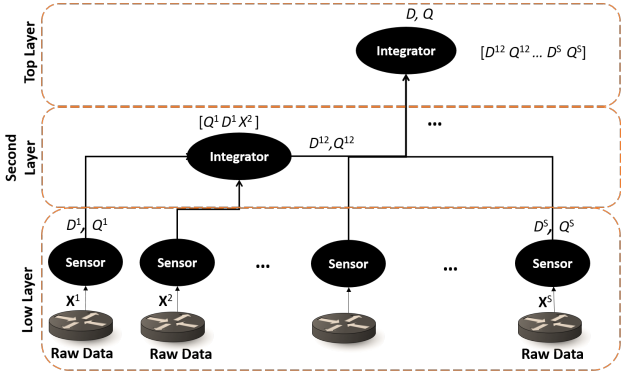
Data Arrangement and Model Building

Three-Mode MSNM. In MSNM, after the parsing and fusion steps, network traffic data are organized in two-way matrices. Recall from Chapter 3 that network data are similar to batch processes if the daily cyclo-stationarity is considered. To do this, the observations in a data matrix $N \times M$ can be re-arranged in a three-way array, where K sampling time points for the $J = M$ monitoring variables are observed during I days[3]. The comparison between the two types of data arrangement is displayed in Fig. 4.8.

[3] We use $I \times J \times K$ instead of $I \times M \times K$ to maintain the same nomenclature as in BMSPC.



(a)



(b)

Fig. 4.7 Examples of hierarchical MSNM (a) with only one integrator and two layers of hierarchy and (b) two integrators and three layers of hierarchy.

The three-way organization of the network data is analogous to the one discussed for a batch process from the process industry, which is described in Section 3.4. Thus, we can use the BMSPC approach to handle the cyclo-stationarity. In this case, the synchronization is not needed, since the days have originally the same length and the main events are expected to be aligned across days. The pre-processing can be performed using TCS (see Chap-

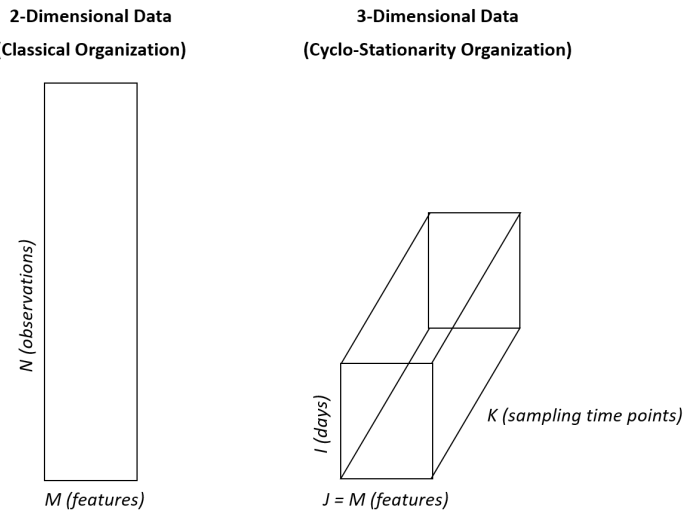


Fig. 4.8 Network Traffic data are typically arranged in two modes (left). An alternative arrangement considers the cyclo-stationarity of a single cycle a day, representing the data as a three-way array (right).

ters 3 and 6). For the unfolding, the batch-wise approach or any form of dynamic model results in a high dimensionality, given the large number of variables typically used in MSNM. Instead, a variable-wise unfolding is recommended. Note that a variable-wise unfolding is actually equivalent to the data distribution that we had prior to the three-way arrangement of the data. Still, the main advantage of conceptually considering the cyclo-stationarity of the data is that most of the dynamics of the data are compensated for in the pre-processing, transforming data from cyclo-stationary to stationary. This is somehow equivalent to go from VCS to TCS (see Chapter 6). This is expected to detect more complex attacks, specially those that are progressively executed along the days and that are not easily detected by the two-way MSNM models. Note that the good properties of the two-way (bilinear) organization of the data are maintained under this approach. The main disadvantage of this approach is that we add uncertainty to the model, since a higher num-

ber of pre-processing parameters need to be estimated from the observations (remember the Parameter Stability Problem discussed in Section 3.4.2).

This extension has been evaluated using real network data in comparison with the standard fusion of the data.

Pre-processing

Recall pre-processing is a step that is needed prior to the construction of the PCA model. Typically, the pre-processing consists on mean centering and, sometimes, Auto-Scaling (AS), so that the same weigh is assigned to each of the variables for the model building.

Optimal Scaling for MSNM. A supervised optimization technique is introduced in [40] to enhance MSNM. This supervised algorithm learns the optimum scaling from the features in the input data. The supervised part of the method optimizes the weights of the features inputted to PCA for the detection of specific attacks. Thus, the system still allows to detect new attacks and, at the same time, it is optimized to detect some attacks of interest. This turns MSNM into a semi-supervised learning approach. The proposed supervised learning is based on an extension of the gradient descent method based on PLS [39, 45]. The objective is to optimize the Area Under the Curve (AUC) of the Tscore by modifying the values of the scaling of the features.

The main advantage of this approach is that the benefits of the unsupervised strategy are maintained, such as the detection of *zero-day* attacks, whilst they are combined with the ability of learning the pattern of a targeted threat, *e.g.* to optimize our detection performance to a current dangerous threat.

PARAmeters from More Observations (PARAMO) for MSNM Based on the three-mode extension of MSNM, PARAMO is an extension for pre-processing that considers the cyclo-stationarity of the data [192]. This ap-

proach is based in the reference method proposed in [149]. PARAMO estimates the means and standard deviations in a three-way tensor using more observations than the original method. This is detailed in Chapter 6 and evaluated using real network data in Chapter 8. The main advantage of PARAMO is that it reduces the uncertainty of the pre-processing parameters and, as a consequence, it increases the capability of fault detection of the monitoring system.

4.2.3 Extensions in the Diagnosis Step

As a part of the MSPC, the root causes of a detected anomaly need to be diagnosed to troubleshoot the problem and/or avoid it in the future. In MSNM, where the number of security events is usually higher than what security operators can handle, the diagnosis takes even a more important role, if possible. The diagnosis helps prioritize the alarms and thus to deal with them in a more effective manner.

Univariate Squared (U-Squared) for MSNM Based in oMEDA [29], the U-Squared considers the full variable space for diagnosis, which results in a univariate diagnosis [195]. It is described in Chapter 7 and evaluated using real network data in Chapter 8. The main advantage of this method is that it enhances the performance of the diagnosis methods because it avoids the *smearing* effect. Being a univariate method, U-Squared simplifies the diagnosis, thus helping the security operators to triage events.

4.2.4 Extensions for Big Data

Recall MSNM has four steps: *1) Parsing*, *2) Fusion*, *3) Detection*, and *4) Diagnosis*. Although originally MSNM was designed as a methodology to deal with Big Data problems [41], the four steps are not enough to deal with

real network data, which volume makes it necessary to extend the original methodology.

The first extension is proposed as a part of a 5 steps methodology [36], which includes an additional step, *Step 5) De-Parsing*. A later work proposes to perform two additional steps to complete the Big Data extension [33]: *Step 0) (Automatic) Feature Extraction* and *Step 6) Visualization Step*. Fig. 4.9 shows the complete integration of these steps. In this figure, the three new steps (which are described in the following paragraphs) are displayed in yellow color, while the original steps from MSNM are represented in black color.

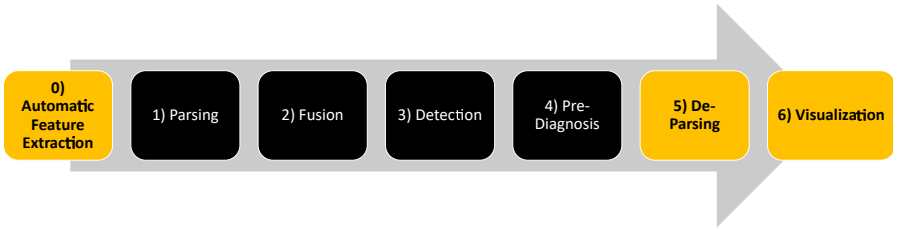


Fig. 4.9 Additional steps for MSNM to deal with Big Data.

De-Parsing Step In network monitoring, after the diagnosis step, it is desirable to come back to the original records to further interpret the diagnosis performed on the monitored variables. The **De-Parsing** consists on identifying the raw information logs related to the anomalies [39]. The information obtained from the *detection* and the *diagnosis* steps is used with this purpose. On the one hand, the detection provides the timestamps for the anomaly, which can be one or a set of consecutive sampling intervals. On the other hand, the diagnosis provides the main variables associated to the anomaly. Crossing timestamps and variables we can identify the raw logs with high accuracy. The complete 5-step methodology is evaluated using real network data in Chapter 8.

Let us consider as an example a botnet attack (from the UGR'16 dataset [130]). The oMEDA diagnosis for the first anomalous observation is shown in Fig. 4.10. This observation corresponds to the 28^{th} of July of 2016.

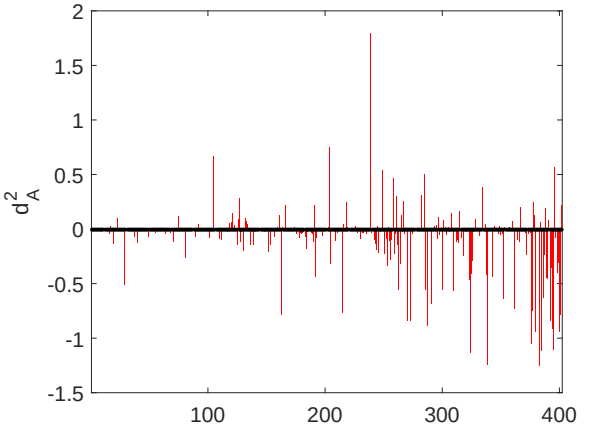


Fig. 4.10 oMEDA diagnosis for the first anomalous observation of the UGR'16.

In our analysis, we observe that the *destination port* `IRC` is the most relevant feature for this anomaly. The `IRC` port is usually configured as 6667. To perform the De-Parsing step, we combine the detection and diagnosis information in a query into the raw data using nfdump on the collected nfcapd, `week5_July`, as follows:

`nfdump -r week5_July ...`
`-t 2016/07/28.00:00:00-2016/07/28.23:59:00...`
`-s ip'dst port 6667'.`

The result of this query confirms that there are several machines receiving a high number of `IRC` packets, with a high traffic flow (see Fig. 4.11 (a)). The De-Parsing step is repeated for a day with no attacks to contrast these results, the 28 of April of 2016, which is also a week day (see Fig. 4.11 (b)). We can

observe that the number of packets, flows and bytes is much higher in July (attack) than in April (NOC).

```
Top 10 IP Addr ordered by flows:
Date first seen          Duration Proto          IP Addr     Flows(%)      Packets(%)       Bytes(%)          pps      bps   bpp
2016-07-28 00:10:35.744 47023.468 any       42.219.154.69      447( 6.5)     1546( 4.3)     79224( 1.5)         0       13    51
2016-07-28 00:00:45.112 47612.060 any       42.219.158.16      377( 5.5)     1476( 4.1)     76120( 1.5)         0       12    51
2016-07-28 00:10:46.384 47012.600 any       42.219.152.20      375( 5.4)     1474( 4.1)     76040( 1.5)         0       12    51
2016-07-28 00:49:52.560 42138.964 any       42.219.156.30      363( 5.3)     1462( 4.1)     75544( 1.5)         0       14    51
2016-07-28 01:00:06.158 20670.718 any       42.219.154.66      292( 4.2)     1391( 3.9)     72416( 1.4)         0       28    52
2016-07-28 01:00:06.158  7166.858 any       42.219.154.70      292( 4.2)     1391( 3.9)     72416( 1.4)         0       80    52
2016-07-28 01:00:06.158  3569.771 any       42.219.154.68      291( 4.2)     1390( 3.9)     72376( 1.4)         0      162    52
2016-07-28 01:00:06.158  3569.771 any       42.219.154.71      291( 4.2)     1390( 3.9)     72376( 1.4)         0      162    52
2016-07-28 01:00:06.158  3569.771 any       42.219.156.27      291( 4.2)     1390( 3.9)     72376( 1.4)         0      162    52
2016-07-28 01:00:06.158  3569.771 any       42.219.156.28      291( 4.2)     1390( 3.9)     72376( 1.4)         0      162    52

IP addresses anonymised
Summary: total flows: 6889, total bytes: 5138218, total packets: 35824, avg bps: 476, avg pps: 0, avg bpp: 143
Time window: 2016-07-27 13:38:48 - 2016-08-08 23:59:58
Total flows processed: 542226955, Blocks skipped: 0, Bytes read: 28195034196
Sys: 16.536s flows/second: 32789127.9 Wall: 16.535s flows/second: 32792074.6
```

(a)

```
Top 10 IP Addr ordered by flows:
Date first seen          Duration Proto          IP Addr     Flows(%)      Packets(%)       Bytes(%)          pps      bps   bpp
2016-04-28 00:12:18.172 85023.116 any     42.219.153.191      111(15.7)      817(15.1)      1.2 M(39.9)         0      115  1500
2016-04-28 00:12:12.532 80455.512 any       42.219.155.28       52( 7.3)      530( 9.8)     173499( 5.6)         0       17   327
2016-04-28 00:02:11.988 80869.492 any     194.231.191.138       38( 5.4)       38( 0.7)     1520( 0.0)         0        0    40
2016-04-28 02:54:20.868 75044.604 any       56.228.69.74       37( 5.2)       37( 0.7)     1480( 0.0)         0        0    40
2016-04-28 00:34:26.496 83273.760 any       56.228.70.238      32( 4.5)       32( 0.6)     1280( 0.0)         0        0    40
2016-04-28 00:10:35.636 82969.384 any      198.132.44.182      29( 4.1)       29( 0.5)     1160( 0.0)         0        0    40
2016-04-28 00:03:25.260 84630.708 any     194.231.140.151      25( 3.5)       25( 0.5)     1000( 0.0)         0        0    40
2016-04-28 00:35:30.472 82647.064 any      198.132.25.203      25( 3.5)       25( 0.5)     1000( 0.0)         0        0    40
2016-04-28 00:05:11.576 81613.716 any      198.132.59.191      23( 3.2)       23( 0.4)      920( 0.0)         0        0    40
2016-04-28 02:08:08.028 72966.860 any       56.228.87.2        21( 3.0)       21( 0.4)      840( 0.0)         0        0    40

IP addresses anonymised
Summary: total flows: 708, total bytes: 3074542, total packets: 5422, avg bps: 285, avg pps: 0, avg bpp: 567
Time window: <unknown>
Total flows processed: 831666755, Blocks skipped: 0, Bytes read: 43244886156
Sys: 23.861s flows/second: 34853196.0 Wall: 23.860s flows/second: 34855588.7
```

(b)

Fig. 4.11 Summary on the raw data after De-parsing (a) anomalous observation (28 July 2016) and (b) NOC observation (28 April 2016). The difference between both days is manifested in the number of flows, packets and bytes.

The 5-step methodology integrates the De-Parsing in an automatic way and provides with free software to make it possible: the FCParser [37].

(Automatic) Feature Derivation Step The volume of real Big Data problems usually is too high to perform feature extraction manually. For this reason, authors in [33] propose an algorithm for self-extracting the features from the data to get a better description of the content. This step needs to be applied prior to the parsing. According to [33], the main characteristics of this learning are: *i)* the main sources of the variance should be captured by the features, and *ii)* characteristics less common should be left as residual

information in the form of "default" features. The latter is built following the original feature-as-a-counter approach. The main advantage of this approach is the automatic consideration of a huge amount of information to extract relevant features in an effective way.

Visualization Step Since De-Parsing also generates a huge amount of raw log entries, the underlying information in such logs is difficult to understand. For this reason, it is suggested to apply a graph visualization technique or tool, such as ***Gephi***[4], turning the results into meaningful and understandable [33].

[4] **Gephi** is a multi-platform Open Source software for data visualization and exploration. Since it allows graph representation (among other features), it is being used in the last years for cybersecurity visualization [19, 135, 202]

Part II

Materials

5

Materials and Methods

"I think you can have a ridiculously enormous and complex dataset, but if you have the right tools and methodology then it's not a problem."
Aaron Koblin, Co-founder and President of Within. Creator and leader of the Data Arts Team at Google from 2008 to 2015

"There are no questions without answer, only bad formulated questions"
Morpheus (1999), The Matrix

Contents

5.1 Implementation Tools 90
5.1.1 MEDA-Toolbox 91
5.1.2 MVBatch Toolbox 94
5.2 Data Generation 95
5.2.1 The UGR'16 Dataset 97
5.2.2 Virtual Network 98
5.2.3 *Saccharomyces cerevisiae* Simulator 98
5.2.4 Synthetic Data 100

This chapter presents the tools used to implement the experimental part of this thesis, as well as the methods utilized for generating the datasets used for the evaluation of such experiments.

5.1 Implementation Tools

All the experiments have been implemented using Matlab®, due to its performance in matrix computation. In addition, two toolboxes have been used throughout this work: The ***MEDA-Toolbox*** (latest update to stable version performed in 2018) [43, 51], which has been utilized in exploratory data analysis, as well as implementing the studied and proposed methods; and ***MVBatch*** (latest update to stable version performed in 2018) [89], which has been mainly used for the batch monitoring experiments performed in Chapter 6.

5.1.1 MEDA-Toolbox

The **MEDA-Toolbox** is a set of multivariate analysis tools for the exploration of datasets [43, 51]. This toolbox also allows dealing with Big Data (in the sense of allowing to deal with unlimited number of observations), which is of high importance for the application of the proposed techniques using real network traffic data.

The MEDA-Toolbox provides classical multivariate exploratory data analysis functionalities, such as score plots or PCA. The MEDA-Toolbox helps to select the number of PCs for PCA and PLS thanks to the application of cross-validation algorithms (take as an example Fig. 5.1 (a)). It also includes more complex tools like MEDA (see Fig. 5.1 (b)) and oMEDA (see Fig. 5.1 (c)), which are intended to study the relationship among variables, and among variables and observations, respectively. The result of MEDA is similar to a correlation matrix, showing the variables that are related grouped in squares. Positive correlations are displayed in red, while negative correlations are shown in blue. The darker is the color, the more correlated the variables are. The result of oMEDA is useful for the diagnosis of one or more observations and it will be detailed in Chapter 7. In addition, the MEDA-Toolbox allows the representation of MSPC charts, including the computation of the statistics and the control limits (see Fig. 5.1 (d)).

There are technical limitations in relation to computation time and memory that make it difficult to manage a dataset when it exceeds a given volume. The Big Data functionality extends the multivariate tools to allow their use without any limitation in the number of observations. Fig. 5.2 shows the same tools displayed in Fig. 5.1 for a Big Data set. The Big Data model building requires the creation of intermediate data structures that are stored to avoid memory overflows. The model can be updated in two ways: *i) Iteratively* and *ii) Following an Exponentially Weighted Moving Average (EWMA)*. Both of them work by clustering similar observations. The main difference is that

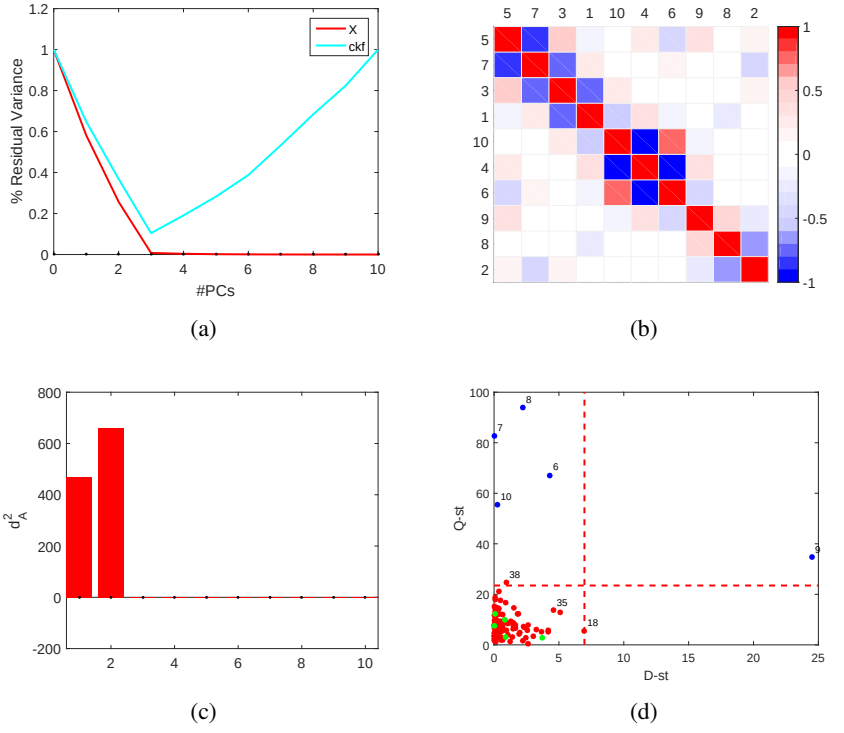


Fig. 5.1 Example of functionalities of the MEDA-Toolbox (a) Cross-validation for selecting the number of PCs (b) MEDA, (c) oMEDA, (d) MSPC.

the **iterative** approach uses all the historical data that are available for the calibration while the **EWMA** approach applies a forgetting factor, λ, to the past observations. λ ranges from 0 to 1, where 0 is the lowest value and means fast adaptation (only present observations are considered to update the model), while 1 is the maximum value and means the consideration of all the past history. In this sense, EWMA enables a better adaptation of the model to the current state of the data. In Chapter 8 we apply the iterative update, since the traffic during the months used for the calibration is expected to be similar and

we think that it is more useful to consider all the past history to create this model.

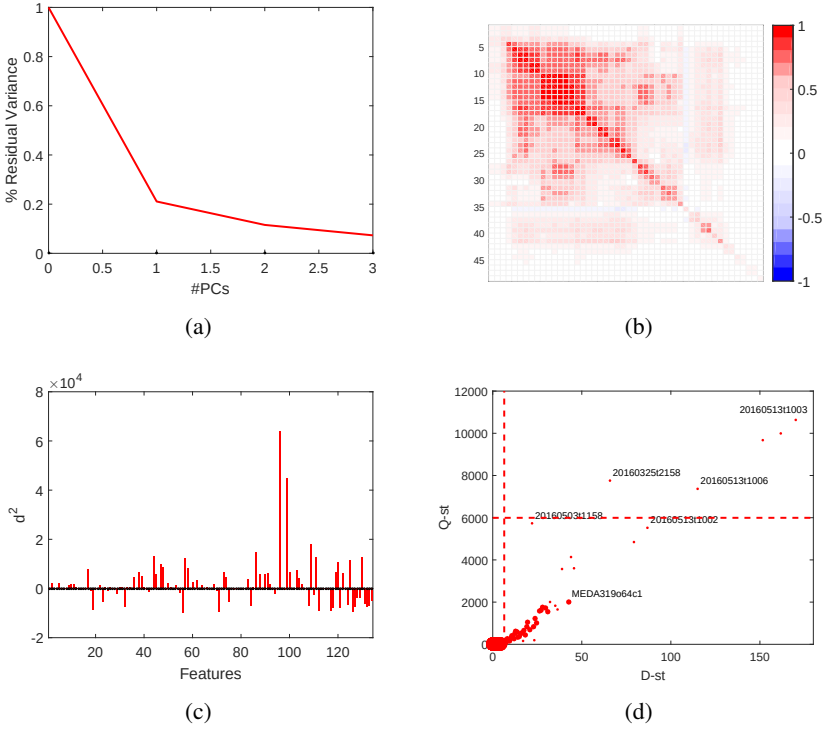
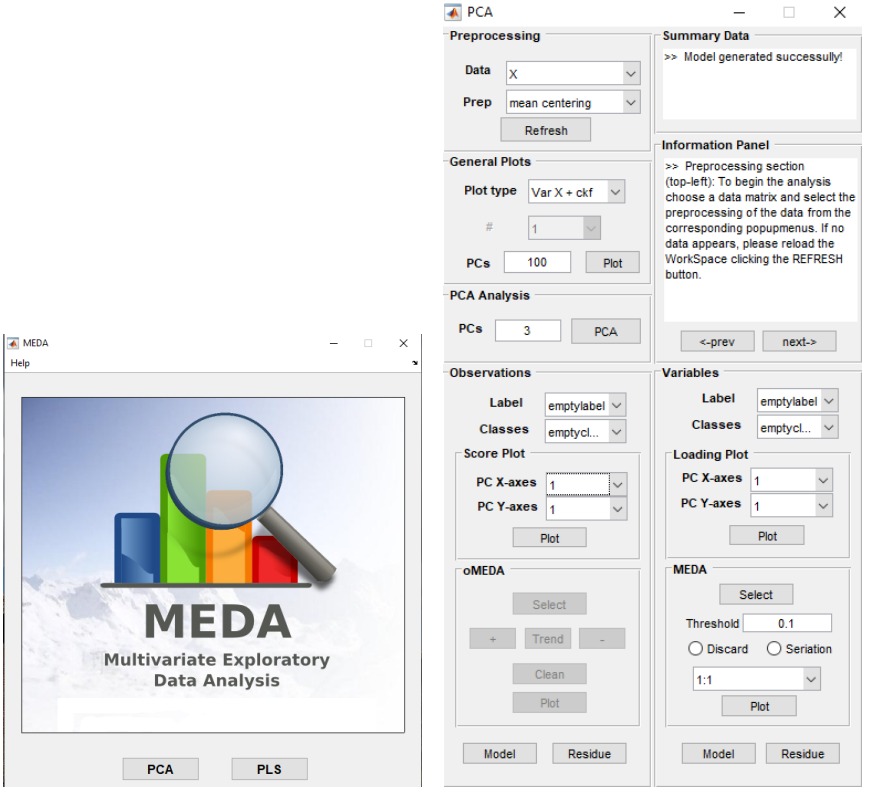


Fig. 5.2 Example of Big Data functionalities of the MEDA-Toolbox (a) PCs selection, (b) MEDA, (c) oMEDA and (d) MSPC (example applying iterative updating).

In addition, the MEDA-Toolbox has a GUI that allows to explore the data. This is specially useful when one starts working with exploratory data analysis, since it is possible to apply either PCA or PLS (see Fig. 5.3 (a)) in combination with MEDA, oMEDA, or any of the implemented EDA methods easily. As an example, Fig. 5.3 (b) shows the GUI for PCA.



(a) (b)

Fig. 5.3 MEDA-Toolbox GUI: (a) Welcome screen allows selecting between PCA and PLS, and (b) Functionalities after selecting the PCA option.

5.1.2 MVBatch Toolbox

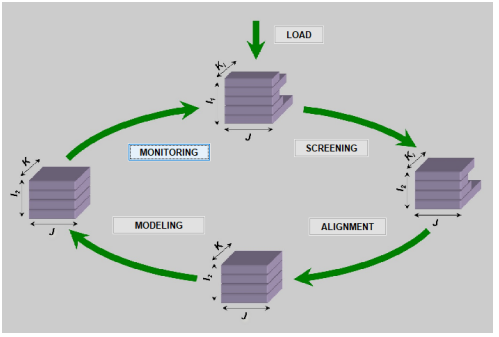
MVBatch is another set of tools for multivariate analysis, which is focused on batch data analysis and monitoring. It implements each of the steps of the batch monitoring cycle: *1) alignment*, *2) pre-processing*, *3) unfolding*, *4) projection* to latent structures, *5) monitoring*, and *6) Diagnosis* [89, 91, 92]. Each of these steps can be performed following both state-of-the-art and

novel algorithms. For example, in Chapter 6, the Relaxed Greedy Time Warping (RGTW) algorithm is applied for the synchronization, TCS for the pre-processing, batch-wise for the unfolding, and PCA for the model fitting. The theoretical control limits are adjusted by using cross validation. The MEDA-Toolbox is integrated in the MVBatch, which also includes a simulator of the *Saccharomyces cerevisiae* cultivation process [89], which is used to generate both NOC and Abnormal Operation Condition (AOC) datasets in Chapter 6.

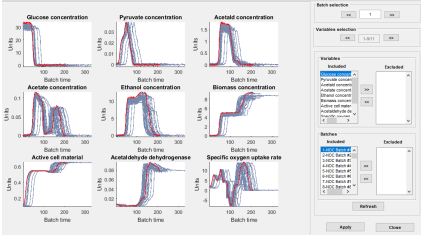
The MVBatch Toolbox has also a GUI that allows to perform the complete batch monitoring cycle, including the visualization of the raw data prior to the synchronization step. The GUI is really useful at the beginning of the experimentation, since it allows the configuration of each of the monitoring steps graphically in an intuitive way. It is possible to select the algorithm to be applied, as well as the input parameters for such algorithm.

5.2 Data Generation

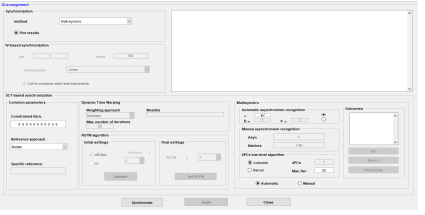
This section describes the mechanisms used for the generation of the main datasets used in the experimentation performed through this PhD. Since this research work has been developed between two research areas, representative datasets originating from each of them have been selected: from cybersecurity, the UGR'16 [130] dataset and data collected from a virtual network [129]; and from the process industry, several datasets are generated using the *Saccharomyces cerevisiae* cultivation process simulator [89, 120]. In addition, and with the purpose of obtaining random data for Monte Carlo simulations, several synthetic datasets are also generated for different correlation levels using the *simuleMV* tool in the MEDA-Toolbox. All of them are described in the following paragraphs.



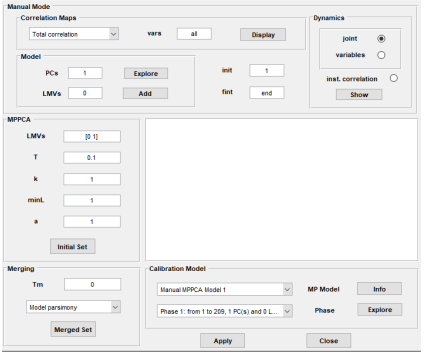
(a)



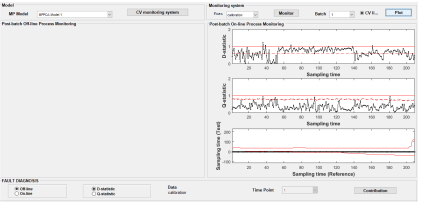
(b)



(c)



(d)



(e)

Fig. 5.4 Example of functionalities across the GUI of the MVBatch Toolbox: (a) Overview Batch monitoring cycle, (b) Visualization of raw data (screening), (c) Alignment, (d) Modeling, and (e) Monitoring.

5.2.1 The UGR'16 Dataset

UGR'16 is a recently published dataset [130] that consists of a collection of `NetFlow` traffic records from a Tier-3 *ISP*. An **Internet Server Provider (ISP)** is a company that provides Internet connection and services to its clients. Some of the most relevant services provided by the ISP from which data were collected are hosting, mail or FTP services. This network hosts a disparate set of clients, which implies an heterogeneous network traffic, making this dataset representative of a wide range of Internet users.

The network traffic was captured during a total of four months. During the fourth month, a series of controlled attacks were launched in the network. Therefore, the capture contains real induced anomalies.

The network traffic was collected by two `NetFlow` sensors configured in two redundant border routers, which provide Internet access to the ISP. There are different sub-networks, one of them hosting the non-protected services, and the other one (internal network) providing firewall protected services to the clients.

To insert the controlled attacks, a total of 25 virtual machines were deployed in some of the sub-networks: 5 attackers referred to as A_1 to A_5, and 20 victims, referred to as V_{11}-V_{45}[1]. Machines A_1 to A_5 attack the rest of the virtual machines (V_i) in different timestamps during a given period of time. Different types of attacks were implemented, and labeled as **DoS**, **scan11**, **scan44** and **botnet**. These attacks are detailed in Section 8.2.2.

All the network traffic is labeled, indicating whether the traffic flow corresponds to: *i) background traffic*, meaning that neither attacks were introduced nor anomalies were detected in the capture; *ii) anomalous traffic*, meaning that anomalies were found in the records; and *iii) attack*, meaning that the flows are artificially induced attacks. Since it is a long capture, one of the

[1] Victim machines belong to four groups. Thus, the name and subscript of the victims have the following structure: $V_{ij}, 1 \leq i \leq 4$ *and* $1 \leq j \leq 5$ [130].

most important features of this dataset is that it allows to detect different types of cycles in the network traffic , *i.e.*, the cyclo-stationarity of the data. As an illustrative example, Fig. 5.5 shows the number of `HTTPS` flows from two different and non-consecutive weeks. The difference between working days (Fig. 5.5 (a)) and weekends (Fig. 5.5 (b)) is evident and daily patterns are also observed.

All the features described in this section make this dataset of main interest for this PhD work, since all the proposals can be validated using real traffic, and this provides an added value our results. More details about UGR'16 can be found in [130].

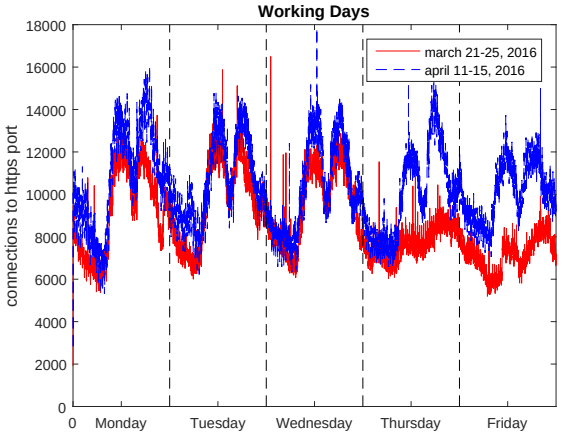
5.2.2 Virtual Network

Network data are collected from the virtual network described in [129]. This virtual network is composed of three routers with 30 client machines in each of them. The information is gathered to a border router and collected during 25 hours. The first 24 hours the network is working under NOC, while the 25th hour includes a number of attacks.

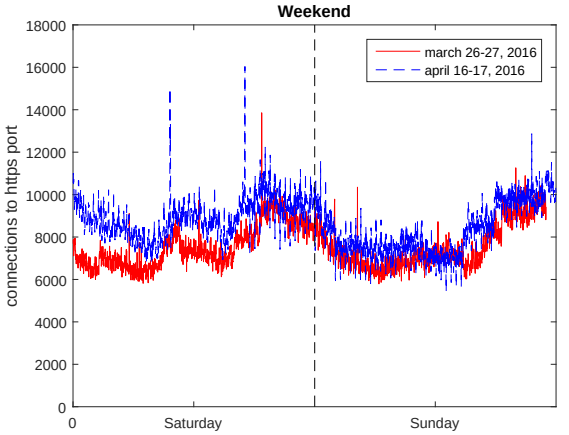
5.2.3 *Saccharomyces cerevisiae* Simulator

In the study of the parallelism between process control and network monitoring, different datasets have been generated using batch process data. The *Saccharomyces cerevisiae* cultivation process simulator has been used to generate several data sets, which are described in detail in Sections 6.6.1 and 7.4.3.

The *Saccharomyces cerevisiae* process simulator is a well known tool that is widely used for the assessment of methods for batch monitoring in the industry [89, 120]. The model of aerobic growth on glucose and ethanol is defined in [120]. In this work, the evolution of different variables (e.g. *ethanol*, *glucose*, *pyruvate* or amount of *biomass* production, among others) is studied taking into account their relationships under different considerations,



(a)



(b)

Fig. 5.5 Connections with destination `HTTPS` port from UGR'16. Differences and similarities between (a) working days and (b) weekends.

namely: *i)* steady-state and *ii)* dynamical conditions. More specifically, the *Saccharomyces cerevisiae* process is modeled under the assumption of glu-

cose limited continuous cultivation, which is characterized by showing two phases of growth: *1)* purely oxidative growth, which produces biomass and carbon dioxide as main products; and *2)* oxido-reductive growth, which mainly produces ethanol because of the alcoholic fermentation caused by glucose repression. During this second growth pyruvate is also generated in different ways as a consequence of the increase in the oxygen uptake rate [120]. The equations that model the reactions corresponding to the *Saccharomyces cerevisiae* cultivation process can also be found in [120].

5.2.4 Synthetic Data

We use **simuleMV** [31] to generate random data with the purpose of performing Monte Carlo experiments. simuleMV allows random data generation if the dimensions of the output matrix are known: the observations, N, and the variables, M. It also allows to fix the correlation level, L. L is defined as an integer ranging from 0 to 9, where 0 means no correlation and 9 means that the correlation is the maximum. By default, the correlation level is 5.

Part III

Contribution to the Multivariate Statistical Network Monitoring

6

Pre-processing

"A person who never made a mistake never tried anything new."
Albert Einstein, Nobel Prize in Physics in 1921

"If you don't like something, change it. If you can't change it, then change your attitude"
Maya Angelou, American writer, poet, singer and civil rights activist

This chapter is mainly based on the following research paper:

- **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019b). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. *Journal of Chemometrics*, 33(11)

Contents

6.1 State-of-the-art Pre-processing Methods 107
6.1.1 The Parameter Stability Problem in the Pre-Processing Context 108
6.2 ROP Enhancement Alternatives 110
6.3 PARAMO 113
6.3.1 *Uniform PARAMO (U-PARAMO)* 113
6.3.2 *eXponential PARAMO (X-PARAMO)* 114
6.3.3 Configuration Values for PARAMO 116
6.4 RADAF 117
6.5 Oversights on the Application of ROP Enhancement Approaches 119
6.5.1 Negative Effects of Asymmetric Windows 119
6.5.2 Negative Effects of RADAF 121
6.6 Materials and Methods 122
6.6.1 Process Control: *Saccharomyces Cerevisiae* . . . 123
6.6.2 Metrics for Evaluation of the pre-processing proposals 127
6.6.3 Finding Comparable Configurations for the Exponential and Uniform Moving Window Methods 129
6.7 Evaluation of the Pre-processing Proposal 130
6.7.1 Results of the Main Experiment 130
6.7.2 Results applying RADAF in model building . . . 139
6.8 Conclusions 143

Data pre-processing is an essential step in MSPC and MSNM. Pre-processing is performed prior to model building, and has a direct effect on

the quality of the calibration model. When one deals with network traffic or process data, it is recommended to center the data to detect deviations around the mean. Most of times it is also needed to homogenize the scale of the variables so that they are given the same relevance in the system. This is often needed since variables frequently come from different sources, and their relative scales are not comparable.

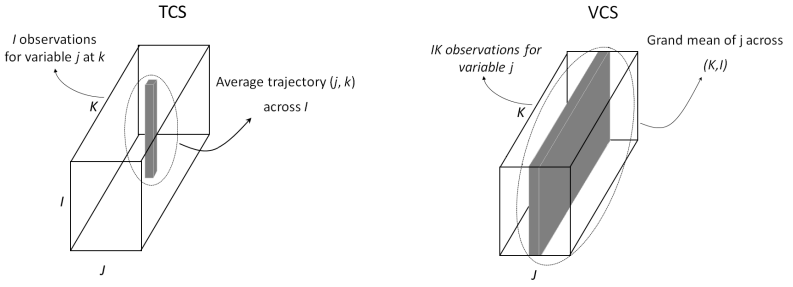
After the parsing and fusion steps in MSNM, network traffic data are normally organized in two-way matrices, where M variables are monitored in N observations over time. To consider the cyclo-stationarity of the data, the observations are re-arranged in a three-way matrix, where K sampling time points for the J monitoring variables are observed during I days[1] (see Section 4.2.2). Taking as a reference the batch monitoring, the most extended pre-processing approaches for three-way arrays are Trajectory Centering and Scaling (TCS) and Variable Centering and Scaling (VCS). The benefit of conceptually considering the cyclo-stationarity of the data is that most of the dynamics of the data are compensated for in the pre-processing, transforming data from cyclo-stationary to stationary. However, as a negative side effect, applying the batch MSPC methods introduces artificial uncertainty in the data [92].

This chapter presents an alternative approach for pre-processing based on TCS. The aim of this approach is to reduce the uncertainty in pre-processing parameters with the final goal of increasing the capability of anomaly detection of the monitoring system. The rest of the chapter describes different variants for this approach and compares them with the reference method, TCS.

[1] Although in the context of MSNM I corresponds to days, in what follows we will talk of batches, for the sake of simplicity.

6.1 State-of-the-art Pre-processing Methods

In batch processes, there are two generally accepted pre-processing methods to consider the differences of magnitude and offsets in variable trajectories: TCS [149] and VCS [214], which conceptual representation is shown in Fig. 6.1. The means are computed in TCS for each variable and sampling time point across all the batches, which is represented as a vertical bar (see Fig. 6.1 (a)). This is termed the **average trajectory** across batches. When VCS is applied, the means are computed for each variable across all the batches and sampling time points, which representation remembers a wall (see Fig. 6.1 (b)). This is termed the **grand mean** of a variable. Equivalent definitions of the standard deviations are used in each approach. Next paragraphs provide the formal definition for both pre-processing methods.



(a) (b)

Fig. 6.1 Visual representation of (a) Trajectory Centering and Scaling (TCS) and (b) Variable Centering and Scaling (VCS) pre-processing.

TCS consists in mean centering and scaling to unit variance the data of each j-th process variable at each k-th sampling time point across I batches (recall from Equation (3.2)). The mean of the variable j at sampling time

point k is computed as follows:

$$\mu_{jk} = \frac{1}{I}\sum_{i=1}^{I} x_{ijk} \tag{6.1}$$

where x_{ijk} is the value of the variable j at sampling time point k in batch i.

The standard deviation of the variable j at sampling time point k is computed from the residuals after subtracting the mean:

$$\sigma_{jk} = \sqrt{\frac{1}{I-1}\sum_{i=1}^{I}(x_{ijk} - \mu_{jk})^2} \tag{6.2}$$

In contrast, VCS performs mean centering and scaling to unit variance on all the data associated with each process variable. The grand mean of each variable j is computed as follows:

$$\mu_j = \frac{1}{IK}\sum_{i=1}^{I}\sum_{k=1}^{K} x_{ijk} \tag{6.3}$$

The standard deviation is computed from the residuals of the corresponding mean as follows:

$$\sigma_j = \sqrt{\frac{1}{IK-1}\sum_{i=1}^{I}\sum_{k=1}^{K}(x_{ijk} - \mu_j)^2} \tag{6.4}$$

where σ_j is the standard deviation of the variable j across the I batches and K sampling time points of the process.

6.1.1 The Parameter Stability Problem in the Pre-Processing Context

In Chapter 3 the parameter stability problem was described. Each of the batch monitoring steps introduce uncertainty, making the model less stable. One

of the main factors affecting the parameter stability is the *Ratio number-of-Observations-to-the-number-of-Parameters (ROP)* [92]. The model parameters (means, standard deviations and loadings) need to be estimated from a number of observations. The **ROP** is the relation between the number of observations that is available to perform the estimation and the number of parameters to be estimated. For a fixed number of parameters to be estimated, the lower the number of observations, the lower the ROP and the higher the uncertainty in the parameters. This makes the parameter stability decrease [92, 192].

TCS only considers I observations to estimate a total of KJ parameters as illustrated in Fig. 6.1 (a), while in VCS the number of parameters is J and the number of observations KI as shown in Fig. 6.1 (b). As a consequence, comparing TCS and VCS, the ROP and thus the parameter stability are much lower in the former than in the latter [92]. A high ROP is an advantage for VCS. However, low uncertainty does not imply that the correct process variation is kept in the residuals after pre-processing [1, 116, 149]. This is illustrated in the following.

Fig. 6.2 represents the pre-processed batch data after applying TCS (left) and VCS (right) on synthetic data [192]. For a better illustration, an anomalous batch (red colored) has been added to the NOC data. This batch is a copy of the tenth NOC batch, in which the values for the sampling time points in the interval $[100, 120]$ are modified to simulate an upset in the process. The anomalous batch is pre-processed using the NOC mean and standard deviation obtained for each pre-processing method. In the example, we can see that TCS is the only pre-processing approach under study that takes the anomalous observations to different (higher) values than the rest of NOC observations, what will enable the fault detection.

In general, TCS is preferred to VCS because the former removes the average trajectory of the batches and focus the BMSPC on the deviations around it (compare Fig. 6.2 (a) with Fig. 6.2 (b) and 6.2 (d)). This makes TCS to be a

more sensible pre-processing approach to detect deviations from normality. On the contrary, VCS removes the grand mean of j across the I batches and K sampling time points, which maintains the systematic variation of the process (compare Fig. 6.2 (a) with Fig. 6.2 (c) and 6.2 (e)). For a thorough discussion on the implications of these pre-processing techniques on modeling and monitoring, the reader is referred to [1, 91, 116, 164, 215]. In what follows we assume TCS as the reference method, in spite of the aforementioned problems of uncertainty of this method. Indeed, these problems are the motivation for next section.

6.2 ROP Enhancement Alternatives

We propose to rearrange the raw data from the 3-way structure in order to increase the number of observations used to calculate the pre-processing parameters, thus increasing the ROP and the parameter stability. Two approaches based in a sliding window scheme are presented to encompass more observations than those considered in TCS for parameter estimation.

- **PARAmeters from More Observations (PARAMO).** This approach calculates the pre-processing parameters for each variable j and time point k across batches by considering a number of neighbored observations in time.
- **RAw DAta Filtering (RADAF).** This approach first filters raw data within the window by considering a number of neighbored observations in time. Then, TCS is applied over the smoothed data.

Fig. 6.3 illustrates the general idea to enhance the parameter estimation. TCS takes I observations to calculate a total of KJ pre-processing parameters (recall from Fig. 6.1 (a)), while the proposed alternative assumes IW observations to estimate these KJ parameters, where W is the number of sampling

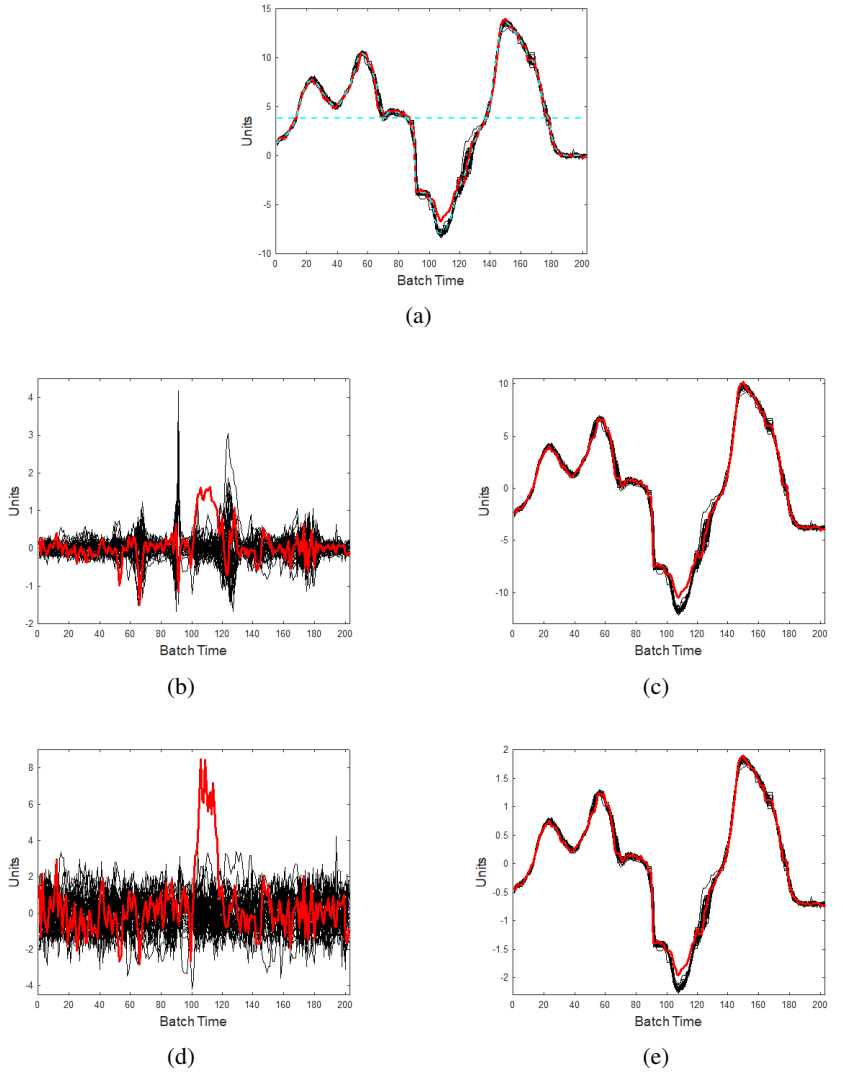


Fig. 6.2 Trajectories for the Specific Oxygen Uptake Rate corresponding to the different batches of the *Saccharomyces Cerevisiae* data set: (a) Raw data with average trajectory and grand mean represented by solid and dashed cyan lines, respectively. Additionally, an anomalous batch has been added, which is displayed with a red solid line. (b) Trajectory centered data, (c) Variable centered data, (d) Trajectory Centered and Scaled data, and (e) Variable Centered and Scaled data.

time points in a given window (see Fig. 6.3). Note that the proposed approach can be considered to be a trade-off between TCS and VCS (compare Fig. 6.3 with Fig. 6.1 (a)) and 6.1 (b)).

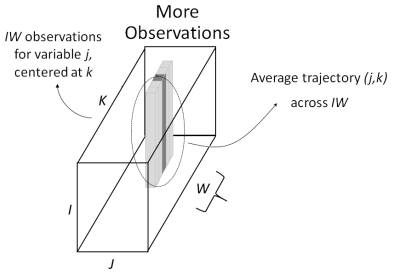


Fig. 6.3 Graphical representation of window-based pre-processing.

On the other hand, two sliding window techniques are used to increase the number of observations:

- ***Uniformly Weighted Moving Window (UWMW)***. The pre-processing parameters are computed treating all observations within the window with equal importance.
- ***Exponentially Weighted Moving Window (EWMW)***. This method weighs observations with a forgetting factor λ, following an exponential function. λ takes values ranging from 0 to 1; the closer to 1, the closer to a uniformly weighted moving window.

Next sections explain how to compute both pre-processing approaches, PARAMO and RADAF, in addition to the corresponding sliding window techniques.

6.3 PARAMO

For the sake of generality, let us consider a window ω_k of size W observations centered at sampling time point k. Later, we will discuss the implications of the application of *asymmetric* windows[2]. The restriction $W = 2L+1, L \in \mathbb{N}^+$ is imposed for W to define the centered window with the same number of preceding and succeeding observations. Also, consider preceding observations are included in the partial window $\overleftarrow{\omega}_k$ whereas succeeding observations are in the partial window $\overrightarrow{\omega}_k$. Thus:

$$\overleftarrow{\omega}_k = \{\mathrm{w}_p, ..., k\} \tag{6.5}$$

$$\overrightarrow{\omega}_k = \{k, ..., \mathrm{w}_s\} \tag{6.6}$$

where w_p and w_s are the extreme values of ω_k and follow:

$$\mathrm{w}_p = max(1, k-L) \tag{6.7}$$

$$\mathrm{w}_s = min(K, k+L) \tag{6.8}$$

6.3.1 *Uniform PARAMO (U-PARAMO)*

Given a three-way data array, $\underline{\mathbf{X}}$, with dimensions I batches, J variables, and K sampling time points, the means and standard deviations are computed following:

$$\mu_k^{\mathtt{UPA}} = \frac{1}{WI} \sum_{w=\mathrm{w}_p}^{\mathrm{w}_s} \sum_{i=1}^{I} \mathbf{x}_{iw} \tag{6.9}$$

[2] An **asymmetric** window is a window that only takes into account observations preceding (or succeeding) the sampling time point k. For the topic under discussion, only an asymmetric window with preceding observations are of interest.

$$\sigma_k^{\mathtt{UPA}} = \sqrt{\frac{1}{WI-1}\sum_{w=\mathrm{w}_p}^{\mathrm{w}_s}\sum_{i=1}^{I}(\mathbf{x}_{iw}-\mu_w)^2} \tag{6.10}$$

where $\mu_k^{\mathtt{UPA}}$ is the vector of mean values and $\sigma_k^{\mathtt{UPA}}$ the vector of standard deviations for all the monitored variables calculated using the measurements spanned by the window ω_k. $\mathbf{x}_{iw}$ refers to the observations of J monitored variables corresponding to the sampling time point w for batch i, and μ_w is the vector of means for the J monitoring variables computed at sampling time point w.

6.3.2 *eXponential PARAMO (X-PARAMO)*

An exponential law [30, 65] is applied in ascending and descending order on each partial window, $\overleftarrow{\omega}_k$ and $\overrightarrow{\omega}_k$. The exponentially weighted moving window estimation of the averages at sampling time point k is computed by averaging the means for preceding and succeeding observations:

$$\mu_k^{\mathtt{XPA}} = \frac{1}{2}\left(\frac{\overleftarrow{\mu}_k}{\overleftarrow{N}_k} + \frac{\overrightarrow{\mu}_k}{\overrightarrow{N}_k}\right) \tag{6.11}$$

where $\mu_k^{\mathtt{XPA}}$ is the array of means for the J monitoring variables computed at sampling time k using the measurements in the window ω_k. $\overleftarrow{\mu}_k$ are the accumulated averages at sampling time point k in the partial window $\overleftarrow{\omega}_k$, and $\overrightarrow{\mu}_k$ are the accumulated averages at sampling time point k in the partial window $\overrightarrow{\omega}_k$. The expressions to estimate the averages in the partial windows are defined as follows:

$$\overleftarrow{\mu}_w = \lambda\overleftarrow{\mu}_{w-1} + \sum_{i=1}^{I}\mathbf{x}_{iw},\ for\ w = \{\mathrm{w}_p,...,k\} \tag{6.12}$$

$$\overrightarrow{\mu}_w = \lambda \overrightarrow{\mu}_{w+1} + \sum_{i=1}^{I} \mathbf{x}_{iw},\ for\ w = \{k, ..., \mathrm{w}_s\} \tag{6.13}$$

Note that $\overleftarrow{\mu}_{w-1} = 0$ when $w = \mathrm{w}_p$ and $\overrightarrow{\mu}_{w+1} = 0$ when $w = \mathrm{w}_s$. $\overleftarrow{N}_k$ and $\overrightarrow{N}_k$ are the number of observations used to calculate the average at sampling time point k in the corresponding partial windows:

$$\overleftarrow{N}_w = \lambda \overleftarrow{N}_{w-1} + I \tag{6.14}$$

$$\overrightarrow{N}_w = \lambda \overrightarrow{N}_{w+1} + I \tag{6.15}$$

with $\overleftarrow{N}_{w-1} = 0$ for $w = \mathrm{w}_p$, and $\overrightarrow{N}_{w+1} = 0$ for $w = \mathrm{w}_s$.

Similar to the means estimation, the array of standard deviations for the J monitored variables at the k-th sampling time point using an exponential function, $\sigma_k^{\mathtt{XPA}}$, is computed using the measurements spanned by the window ω_k as follows:

$$\sigma_k^{\mathtt{XPA}} = \frac{1}{2}\left(\sqrt{\frac{1}{\overleftarrow{N}_k - 1}\overleftarrow{\sigma}_k^2} + \sqrt{\frac{1}{\overrightarrow{N}_k - 1}\overrightarrow{\sigma}_k^2}\right) \tag{6.16}$$

where $\overleftarrow{\sigma}_w^2$ and $\overrightarrow{\sigma}_w^2$ are the accumulated variances at sampling time point k. These accumulated variances are calculated removing the corresponding average at each sampling time point w in the partial window:

$$\overleftarrow{\sigma}_w^2 = \lambda \overleftarrow{\sigma}_{w-1}^2 + \sum_{i=1}^{I} (\mathbf{x}_{iw} - \overleftarrow{\mu}_w)^2,\ for\ w = \{\mathrm{w}_p, ..., k\} \tag{6.17}$$

$$\overrightarrow{\sigma}_w^2 = \lambda \overrightarrow{\sigma}_{w+1}^2 + \sum_{i=1}^{I} (\mathbf{x}_{iw} - \overrightarrow{\mu}_w)^2,\ for\ w = \{k, ..., \mathrm{w}_s\} \tag{6.18}$$

with $\overleftarrow{\sigma}_{w-1}^2 = 0$ for $w = \mathrm{w}_p$, $\overrightarrow{\sigma}_{w+1}^2 = 0$ for $w = \mathrm{w}_s$.

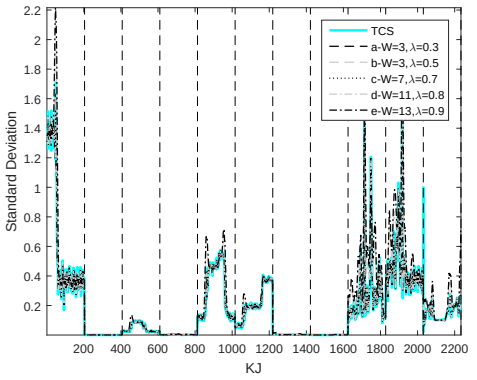
6.3.3 Configuration Values for PARAMO

The size of window, W, needs to be selected for PARAMO. In the case of X-PARAMO, we need to set also λ. However, not all the possible values will produce accurate results. A large size of the window (and λ) produces an excess of smoothing in the pre-processing parameters, which can lead to a relevant information loss. This excessive smoothing is graphically observed in the form of artifacts. These artifacts are specially visible in the trajectories for the standard deviations. Fig. 6.4 shows a practical example of this effect, computed for the first process variable of a NOC data set of the *Saccharomyces Cerevisiae* cultivation process [89] (see Chapter 5) where TCS is displayed in cyan and X-PARAMO in gray. X-PARAMO is applied for different sizes of W and values of λ. For $W < 11$, there are no artifacts, while for the greater sizes there is an increment in the magnitude of the standard deviation.

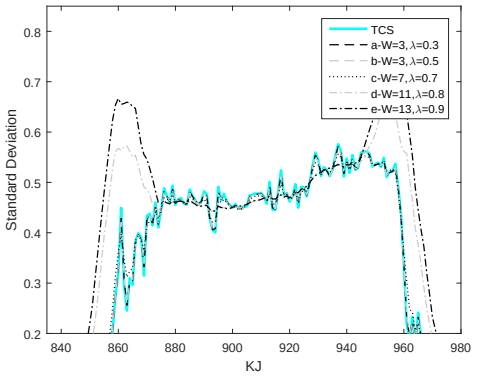
The presence of artifacts can also be mathematically quantified, by computing the *Sum of Squared Residuals (SSR)* as described in the following:

$$SSR_m = \sum_{j=1}^{J} \sum_{k=1}^{K} (\sigma_{k,j}^{TCS} - \sigma_{k,j}^{m})^2 \tag{6.19}$$

where m is the moving window method. The SSR is computed between the trajectory of the pre-processing parameters computed i) by TCS and ii) applying the moving window method with the selected configuration for the chosen ROP increasing approach. The SSR allows the comparison of different pre-processing approaches.



(a)



(b)

Fig. 6.4 Effect of applying different levels smoothing on (a) the trajectory of the standard deviation across multiple variables and (b) zoom over the 5^{th} process variable of the *Saccharomyces cerevisiae* process cultivation.

6.4 RADAF

Let us consider the same restrictions and extreme time points for ω_k as defined in Equations (6.7) and (6.8).

Uniform RADAF (U-RADAF)

Given a three-way data array, $\underline{\mathbf{X}}$, with dimensions I batches, J variables, and K sampling time points, the smoothing of the RADAF approach is performed as follows:

$$\tilde{\mathbf{X}}_k^{\mathtt{URA}} = \frac{1}{W} \sum_{w=\mathrm{w}_p}^{\mathrm{w}_s} \mathbf{X}_w \tag{6.20}$$

where $\tilde{\mathbf{X}}_k^{\mathtt{URA}}$ corresponds to the filtered data of all the process variables and batches computed using the measurements spanned by the window of size W centered at the sampling time point k. $\mathbf{X}_w$ is the two-way array extracted from $\underline{\mathbf{X}}$ at sampling time point w. Means and standard deviations are computed from $\tilde{\mathbf{X}}_k^{\mathtt{URA}}$ using TCS.

eXponential RADAF (X-RADAF)

In RADAF, like in PARAMO, the window ω_k is split into two partial windows, $\overleftarrow{\omega}_k$ and $\overrightarrow{\omega}_k$. The exponential law for RADAF is recursively calculated for each partial window as follows:

$$\overleftarrow{\tilde{\mathbf{X}}}_w = \lambda \overleftarrow{\tilde{\mathbf{X}}}_{w-1} + (1-\lambda)\mathbf{X}_w, w = \{\mathrm{w}_p, ..., k\} \tag{6.21}$$

$$\overrightarrow{\tilde{\mathbf{X}}}_w = \lambda \overrightarrow{\tilde{\mathbf{X}}}_{w+1} + (1-\lambda)\mathbf{X}_w, w = \{k, ..., \mathrm{w}_s\} \tag{6.22}$$

$\overleftarrow{\tilde{\mathbf{X}}}_w$ and $\overrightarrow{\tilde{\mathbf{X}}}_w$ are the filtered observations corresponding to preceding and succeeding windows, respectively.

The filtered value of $\mathbf{X}_k$ (raw data at sampling time point k) is computed as:

$$\tilde{\mathbf{X}}_k^{\mathtt{XRA}} = \frac{1}{2}(\overleftarrow{\tilde{\mathbf{X}}}_k + \overrightarrow{\tilde{\mathbf{X}}}_k) \tag{6.23}$$

where $\tilde{\mathbf{X}}_k^{\mathtt{XRA}}$ corresponds to the filtered data for all the process variables and batches computed at sampling time point k. Means and standard deviations are computed from $\tilde{\mathbf{X}}_k^{\mathtt{XRA}}$ using TCS.

Next section describes some pitfalls when the pre-processing enhancement is applied.

6.5 Oversights on the Application of ROP Enhancement Approaches

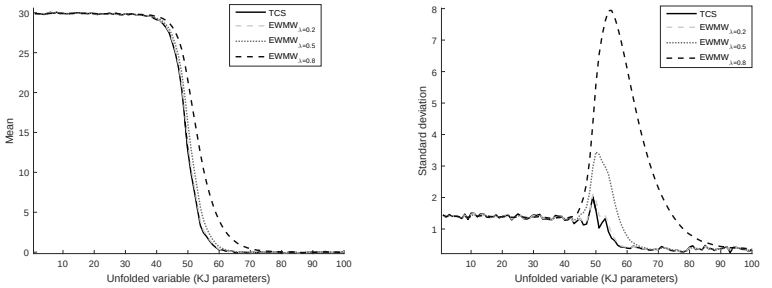
6.5.1 Negative Effects of Asymmetric Windows

The application of the proposed pre-processing variants in an asymmetric window, which only includes preceding observations to sampling time point k, causes artifacts both on RADAF and PARAMO.

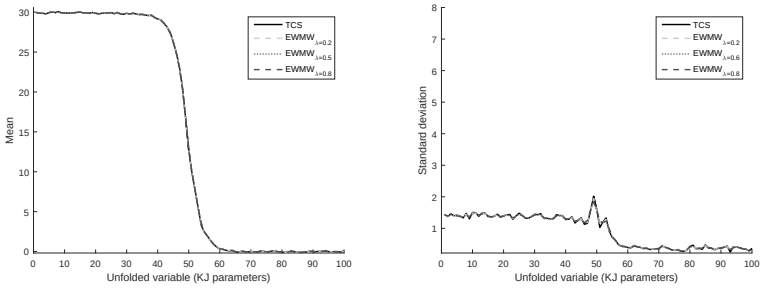
Let us take as for illustration the pre-processing proposal X-PARAMO. Fig. 6.5 (a) shows the means and standard deviations computed for the first process variable of a NOC data set of the *Saccharomyces Cerevisiae* cultivation process (see Section 6.6.1 for data Generation). The pre-processing parameters have been computed by TCS and by X-PARAMO applying an asymmetric window. When computing the average trajectory, an exponential shift related to λ is artificially created. This in turns creates exponential artifacts (peaks) in the standard deviation.

However, when the window is centered on the sampling time of interest, none of these artifacts are generated. Fig. 6.5 (b) shows the result of applying X-PARAMO in a centered sliding window. The shift appearing in the pre-processing parameters when an asymmetric smoothing applied is corrected. The mean and standard deviation trajectories for different values of λ almost overlap with the resulting trajectories for TCS (compare black solid line with dashed lines in Fig. 6.5 (b)). The main difference lies in the smoothing effect when a high λ is applied. In addition, the peaks related to the artificial variation introduced in the estimates are drastically diminished (see right graph in Fig. 6.5 (b)). Note that the smoothing is more prominent in the standard

deviation than in the averages because of the propagation effect. Similar effects are also observed for the application of RADAF in a centered window.



(a) Means (left) and standard deviations (right) calculated for the first process variable applying PARAMO and the **asymmetric** exponential law.



(b) Means (left) and standard deviations (right) calculated for the first process variable applying PARAMO and the **symmetric** exponential law.

Fig. 6.5 Comparison of the pre-processing parameters (means and standard deviations) computed for the first process variable of the *Saccharomyces cerevisiae* process cultivation with TCS and PARAMO based on (a) asymmetric EWMW and (b) symmetric EWMW

.

To mitigate the addition of spurious variability by asymmetric methodologies, we recommend the application of symmetric moving window methods for both RADAF and PARAMO. Note that applying a centered window is possible since the model is built in an off-line mode from historical observa-

tions, and thus, it does not introduce any delay in the monitoring (exception made on the discussion in next sub-section).

6.5.2 Negative Effects of RADAF

Fig. 6.6 (a) illustrates a limitation of RADAF: the standard deviation is underestimated, as observed in the last period of the batch. The reason for this effect is the filtering performed by RADAF, which reduces the uncertainty in the estimation of the mean and therefore the variance associated. This effect is better observed in steady periods of the average trajectory. While this is not a problem or disadvantage by itself, it means that new batches under real-time monitoring will need the same filtering as in the calibration data. According to Section 6.5.1, windows need to be centered to avoid artifacts, which means that the proper application of RADAF in the monitoring phase would cause a delay of $\lfloor W/2 \rfloor$ sampling time points.

Taking into account Sections 6.5.1 and 6.5.2, our general recommendation is to use PARAMO with the *symmetric* application of UWMW (U-PARAMO) or EWMW (X-PARAMO). However, RADAF is not discarded on the experimental work, so that we are able to evaluate and compare the effect of its application.

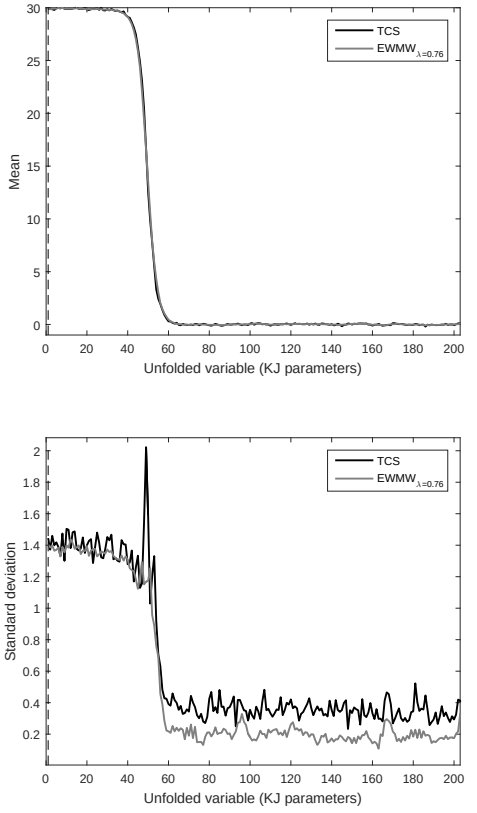


Fig. 6.6 Comparison of the pre-processing parameters (means and standard deviations) computed for the first process variable of the *Saccharomyces cerevisiae* process cultivation with RADAF following the symmetric exponential law (X-RADAF) and TCS.

6.6 Materials and Methods

The aim of this section is to present the datasets and metrics needed to compare the pre-processing proposals, and it is guided by three questions:

1. Which is the best enhancing approach, PARAMO or RADAF?

2. Which is the best moving window method, exponential (X) or uniform (U)?

3. Does the best pre-processing approach of our proposal outperform TCS?

RADAF has been included in the experiments to complement the study of the problems described in Section 6.5.2. In the main experiment, RADAF is applied only to obtain the pre-processing parameters during the calibration. Then, the new test data are pre-processed without any filtering. The aim is to assess whether applying RADAF only to estimate the pre-processing parameters is a valid approach for on-line monitoring. However, in network monitoring (and probably in some other fields), a certain delay could be affordable if the detection results are improved. For this reason, a second experiment has been performed to complete the study with the evaluation of RADAF applied also to the model building and the test data. Thus, both calibration and test data are filtered with a symmetric window, introducing a delay of $\lfloor W/2 \rfloor$ in on-line monitoring. Corresponding results for this second experiment are shown in Sub-section 6.7.2. Note that PARAMO only acts on pre-processing parameters.

6.6.1 Process Control: *Saccharomyces Cerevisiae*

Given the analogy between MSPC and MSNM, one of the selected datasets for the experimental part of this thesis comes from the process control area. This is a well-known process benchmark, the *Saccharomyces Cerevisiae* cultivation, which is described in the next paragraphs. These data are generated using the simulator of the fermentation process [89] and are used for the comparison of the pre-processing approaches and monitoring systems. The simulation of NOC batch data is based on the stoichiometric biological model published in [120]. In addition, Gaussian noise of low magnitude is added to the initial conditions (10%) and to batch data (5%).

For the simulation of Abnormal Operation Condition (AOC) batch data, the aforementioned parameters are altered, producing an anomaly that affects the normal metabolic behavior described by the stoichiometric reactions.

A total of five types of datasets are simulated for this research study both for NOC and AOC. All the datasets contain $J = 11$ process variables (see Table 6.1):

Dataset type	*#Datasets*	*#Batches*	*Target*
NOC	100	30	Parameter stability assessment
NOC	1	25	Model validation
FI: *k1l* modified, $t \in [1,35]$	50	20	Fault detection assessment
FII: *k6* modified, $t \in [1,15]$	50	20	Fault detection assessment
FIII: *k6* modified, $t \in]15,35]$	50	20	Fault detection assessment

Table 6.1 Synthetic datasets generated with the fermentation process of the *Saccharomyces Cerevisiae* cultivation. *k1l* and *k6* are two of the kinetic parameters defined in [120], which were modified to generate FI, FII and FIII.

For the sake of creating realistic process upsets, a gradual deterioration is imposed in the simulation: the kinetic parameters gradually vary in each iteration to simulate a deviation from normality across batches, and therefore, over production time. This implies that the first batches will not completely reflect the fault as much as the last batches. The aim is to mimic real abnormal scenarios in industrial processes, such as fouling, catalyst deactivation, and product quality degradation, and illustrate the difficulties to detect early subtle drifts with monitoring systems.

After the data generation, for each dataset (NOC and AOC), all the batches are synchronized against a NOC batch whose duration is the median of the batch length of the first dataset. In this case, the RGTW algorithm is used [90, 92]. No constraints in variables are imposed and the synchronization parameters are adjusted by following a cross-validation approach [90].

The results of the simulation for each fault are shown in Fig. 6.7. The overall Q-statistic values computed for each batch and type of fault show a

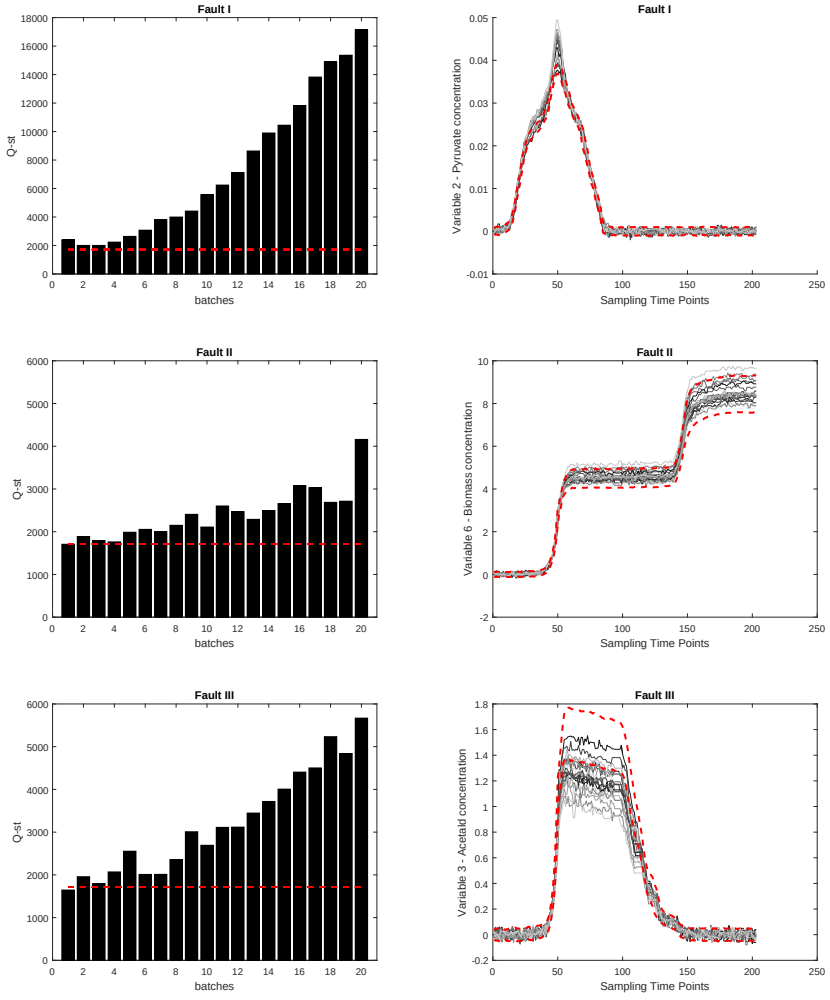


Fig. 6.7 Synthetic faults generated from the simulated fermentation process of the *Saccharomyces cerevisiae* cultivation: evolution of the overall Q-statistic over batches for three different faults, with control limits depicted as red dashed lines (left), and raw batch trajectories of the variables involved in the failure with NOC ranges shown with red dashed lines (right).

gradual increase in the magnitude of the fault (see Fig. 6.7 (a)). The degradation of the operating conditions over time is represented by a gradient gray color. The clearer the color, the longer the time the process has been affected by the abnormality. The band of normal operation conditions for each variable is also shown in red dashed lines to ease the identification of the fault.

1. 100 NOC datasets are simulated for comparison purposes in terms of parameter stability, with 30 batches each.

2. A single independent set is generated with 25 batches for validation purposes. These batches are used to homogenize the accuracy of the final monitoring systems in terms of the Overall Type I (OTI) error.

3. Three different faults are simulated to evaluate the performance of the monitoring systems in terms of Overall Type II (OTII) error. A total of 50 independent datasets, with 20 abnormal batches each, are created for the three process disturbances. Each abnormality affects a different metabolic route of the yeast[3]. The features of the simulations performed for each fault are:

    - **Fault I**. Constant $k1l$ is gradually modified across batches in the fermentation time interval $1h \leq t \leq 35h$ to induce an abnormality in the glucose uptake system and the glycolytic pathway. The resulting abnormality affects the entire batch run.
    - **Fault II**. Constant $k6$ is gradually modified across batches in the fermentation time interval $1h \leq t \leq 15h$ to induce a fault associated with an abnormal formation of ethanol from acetaldehyde. The abnormality affects the lag phase, in which the yeast becomes acclimated to the heterogeneous culture media. As a result, the yeast is shifted from dormancy to metabolic activity.

[3] Two of the kinetic parameters (constants) defined in [120] are modified to generate the faults. These constants are: $k1l$ and $k6$.

- **Fault III**. Same fault as Fault II, with the difference that the disturbance is simulated in the second half of the process ($15h \leq t \leq 35h$), affecting the other fermentation phases: the first and second exponential growth, and the stationary phase.

6.6.2 Metrics for Evaluation of the pre-processing proposals

Parameter Stability Assessment

The effects on the parameter stability are studied through the *Normalized Squared Difference (NSD)*, which is conducted between pairs of NOC datasets to compare and evaluate the parameter stability of the pre-processing methods. For the present work, this value is computed for each of the model parameters, θ, which is used to represent the means, standard deviations and loadings. θ is distributed in two independent and identically distributed samples as described in [92]:

$$NSD_{\theta} = \frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{J}\left(\frac{\theta_{jk}^{(1)}}{\|\theta_{k}^{(1)}\|} - \frac{\theta_{jk}^{(2)}}{\|\theta_{k}^{(2)}\|}\right)^2 \tag{6.24}$$

where $\theta_{jk}^{(1)}$ and $\theta_{jk}^{(2)}$ are the parameter values corresponding to the j-th variable at sampling time point k in the model parameter vectors $\theta^{(1)}$ and $\theta^{(2)}$ for the first and second datasets under comparison, respectively. $\theta_{k}^{(1)}$ and $\theta_{k}^{(2)}$ are the parameter arrays corresponding to sampling time point k. To estimate the NSD for the loadings the sign of each loading vector is corrected by the sign of the absolute maximum loading value.

The lower the NSD value, the lower the uncertainty in the evaluated parameter, and thus the higher the stability of such parameter [92].

Performance of monitoring systems

In this part of the section, the metrics for the accuracy of the models to detect anomalies are described.

The **OTI** is used to compute the percentage of false positives, *i.e.* detected anomalies that do not contain any security problem or fault. It is used to normalize different monitoring systems [44, 95]. Another important measure is the number of false negatives, corresponding to real anomalies that are not detected. The corresponding metric is the **OTII** [49, 162, 203].

The OTI risk values are computed as follows:

$$OTI = 100\frac{\#f}{K \cdot I} \tag{6.25}$$

where $\#f$ refers to the number of false positives. The OTI should be as close as possible to the imposed significant level in the UCL [44, 95], which is of special relevance in the case of cybersecurity.

The OTII is calculated as follows:

$$OTII = 100\frac{\#nf}{\sum_{i=1}^{I_{faulty}} l(i)} \tag{6.26}$$

where $\#nf$ is the number of false negatives, I_{faulty} the number of faulty batches, and $l(i)$ the length of the abnormality in each batch (true faults). The OTII value should also be as close to 0 as possible.

The OTI risk values are computed from an independent set of NOC batches (I_{NOC}) not used for model building, using Equation 6.25. Following [44, 95], we set the control limits of all monitoring systems with different pre-processing methods so that the OTI value is close to the imposed significance level α, in order to enable a fair statistical comparison between modeling approaches. To study the impact of enhanced parameter stability on process monitoring, the accuracy of the monitoring systems needs to be evaluated

for the different associated pre-processing approaches. For such purpose, the percentage of faults not detected are calculated following Equation 6.26.

The three questions at the beginning of the section are assessed in terms of the NSD value to evaluate the parameter stability [92] and, in terms of the OTII value to assess the accuracy of the monitoring systems to detect faults [49, 162, 203]. As a first step, the synchronized batch data are pre-processed using the methods under study. Afterwards, the resulting data are batch-wise unfolded. Finally, PCA is applied for model building. The first PC is selected to construct the model:

- For the first study, to evaluate parameter stability, the NSD values are computed for each consecutive pair of simulated NOC datasets yielding 50 values per combination of approach and moving window method, for each pre-processing parameter.
- For the second study, to evaluate monitoring performance, a monitoring system is designed. After the model building, the control limits of the monitoring system are adjusted by cross-validation. The performance of the monitoring systems is subsequently assessed for the three faults generated (Fault I, II and III). This study implies the statistical analysis of 50 OTII values, as many as datasets generated per fault. The OTII are computed following Equation 6.26, where the false negatives are the total number of non-detected faults neither by the D-statistic nor by the Q-statistic.

6.6.3 Finding Comparable Configurations for the Exponential and Uniform Moving Window Methods

All pre-processing variants should also be homogenized in terms of the smoothing caused in the trajectories of pre-processing parameters due to the application of the sliding windows. We consider that the smoothing effect of the

uniform window versions of RADAF and PARAMO are equivalent for the same window size. Then, we set the level of smoothness of the exponential versions to be as close as possible to the corresponding uniform versions. This is illustrated in Fig. 6.8 for PARAMO. Potential values of λ and W for X-PARAMO are evaluated in terms of the SSR (see Equation (6.19)) and compared to SSR_0, which represents U-PARAMO. The closest setting to SSR_0 is chosen. The range of settings S for the homogenization of X-PARAMO with U-PARAMO is $0 \leq \lambda < 1$ and $W \in [3,5,7,9,11]$. The procedure is run for each of the 100 NOC data sets, considering two different window widths for the reference method: $W_{ref} \in [3,5]$.

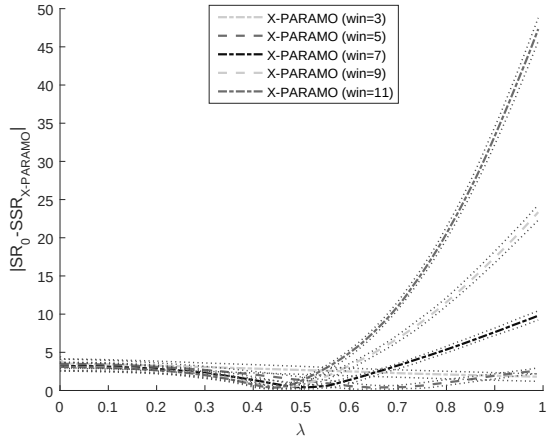
Fig. 6.8 shows the smoothing homogenization procedure of X-PARAMO for U-PARAMO with $W = 3$ (Fig. 6.8 (a)) and $W = 5$ (Fig. 6.8 (b)). In both cases, median values (plus quartiles) for the 100 random repetitions are shown. We look for the minimum values in the curves and, in case of similar results, we choose the lowest window size.

Let us take as an example the homogenization for U-PARAMO with a reference window of size $W = 5$, which is shown in Fig. 6.8 (b). X-PARAMO with window sizes $W = 7$, $W = 9$, and $W = 11$ present similar values of d. Here we recommend to select the configuration for the exponential weighting $W = 7$ and $\lambda = 0.89$, choosing the minimum window size. Equivalently, for U-PARAMO with $W = 3$ we would choose X-PARAMO with $W = 7$ and $\lambda = 0.51$.

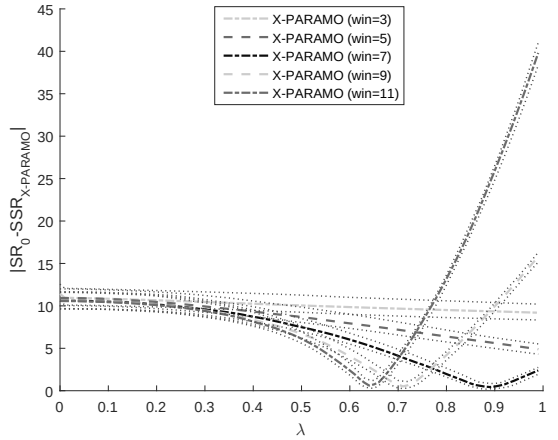
6.7 Evaluation of the Pre-processing Proposal

6.7.1 Results of the Main Experiment

We study the parameter stability (NSD) and monitoring performance (OTII) using a designed experiment with three factors: the moving window method,



(a) Reference window $W = 3$.



(b) Reference window $W = 5$.

Fig. 6.8 Homogenization of the configuration values of X-PARAMO to the reference method, U-PARAMO, with uniform size (a) $W = 3$ and (b) $W = 5$.

the level of smoothness, and the pre-processing approach. Results are analyzed using ANalysis Of VAriance (ANOVA) as described in Table 6.2.

Factor	*Level*	*Description*
$f1$	U	Moving Window Method
	X	
$f2$	W1	Setting
	W2	
$f3$	RADAF	Approach
	PARAMO	
$f12$		Interaction $f1$ and $f2$
$f13$		Interaction $f1$ and $f3$
$f23$		Interaction $f2$ and $f3$
$f123$		Interaction $f1$, $f2$ and $f3$

Table 6.2 Factors, levels and interactions considered for the 3-factor ANOVA studies. U represents the uniform and X the exponential moving window method. W1 and W2 are the configuration settings corresponding to W=3 and W=5 for uniform windows and their corresponding homogenized values for exponential windows (see Section 6.6.3), respectively.

Which is the best enhancing approach and moving window method for parameter stability?

One separated ANOVA is performed on the NSD values[4] for each parameter vector (means, standard deviations and loadings) to assess whether the effects on parameter stability are statistically significant across pre-processing methods. Additionally, the effect size $\eta = SS(f)/SS(total)$ is employed to identify the most relevant factors (see Table 6.3).

NSD	$f1$	$f2$	$f3$	$f12$	$f13$	$f23$	$f123$
Means	✗	✓($\eta = 0.0132$)	✗	✗	✗	✗	✗
Standard Deviations	✓($\eta = 0.0113$)	✓($\eta = 0.0172$)	✓($\eta = 0.2803$)	✗	✗	✓($\eta = 0.0113$)	✗
Loadings (P#1)	✗	✓($\eta = 0.0241$)	✓($\eta = 0.4558$)	✗	✗	✓($\eta = 0.0240$)	✗

Table 6.3 Results of the 3-Factor ANOVAs for the parameter stability evaluation. ✓indicates statistical significant differences ($p - value < 0.05$). ✗ indicates absence of statistical significant differences.

[4] Given the positive skewness of the NSD values, a logarithmic transformation is applied prior to ANOVA.

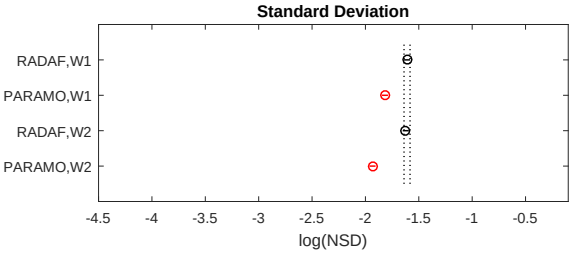
The interaction between $f2$ and $f3$ is statistically significant ($p-value < 0.05$) for the standard deviations and the loadings. All the main effects are statistically significant for the standard deviations. In the case of the loadings, all the factors except for $f1$ are also statistically significant. However, for the means the only relevant factor is $f2$ (the setting). We believe this is due to the fact that means are more stable than the rest of the parameters, given the propagation of the uncertainty from means to standard deviations and to the loadings. Thus, the effect of applying PARAMO or RADAF is not evidenced on the means.

The Least Significant Difference (LSD) intervals for the interactions between $f2$ and $f3$ are shown in Fig. 6.9 and reflect that PARAMO outperforms RADAF for the standard deviations and loadings and for the settings under study. According to η, the most relevant factor is $f3$ (PARAMO vs RADAF). The NSD values are lower for PARAMO than for RADAF, which means that PARAMO yields more stable pre-processing parameters. Thus, we select PARAMO for the rest of the parameter stability study. The LSDs represented in Fig. 6.11 show that, for the standard deviations, the exponential (X) window method is significant better in the NSD values than the uniform (U) method.

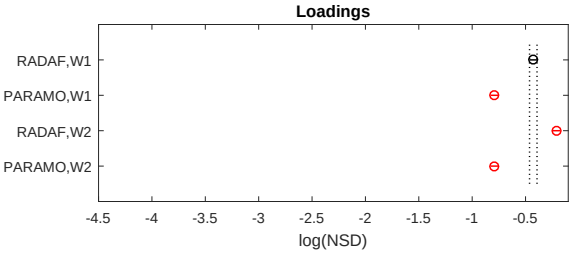
Which is the best enhancing approach and moving window method for monitoring?

The ANOVA scheme in Table 6.2 is repeated on the OTII values for each of the three generated faults. The goal is to study the differences on accuracy of the models to detect faults. The results are displayed in Table 6.4.

The resulting ANOVAs show a statistical significant difference in the interaction between $f1$, $f2$, and $f3$ (denoted as $f123$) only for Fault II ($p-value < 0.05$). The corresponding LSD intervals for these interactions displayed in Fig. 6.12 show that PARAMO present better results than RADAF.



(a) Standard Deviations.



(b) Loadings.

Fig. 6.9 LSD intervals at 95% confidence level of the NSD values for the interactions between enhancing approach PARAMO and RADAF and the selected configuration settings W1 and W2 computed on (a) the standard deviations and (b) the loadings.

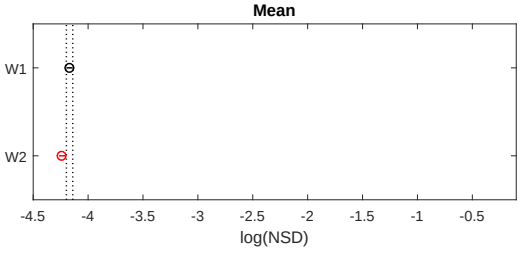


Fig. 6.10 LSD intervals at 95% confidence level of the NSD values for the settings W1 and W2 computed on the means from the 3-factor ANOVA.

$f1$ and $f3$ and their interaction present statistical significant differences in Fault I and Fault III. The LSD intervals corresponding to the interactions between $f1$ and $f3$ in Faults I and III are depicted in Fig. 6.13, and show

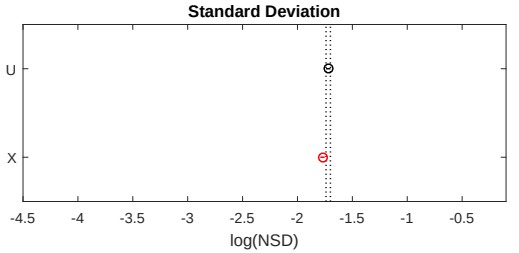


Fig. 6.11 LSD intervals at 95% confidence level of the NSD values for moving window methods Uniform and eXponential computed on the standard deviations from the 3-factor ANOVA.

that the false negatives, or OTII values, are generally lower on average for PARAMO than RADAF. The rest of the experiments are performed applying PARAMO. In addition, we can observe that the OTII values are generally lower in X-PARAMO than in U-PARAMO, regardless of the simulated fault.

OTII	$f1$	$f2$	$f3$	$f12$	$f13$	$f23$	$f123$
Fault I	✓($\eta = 0.0066$)	✗	✓($\eta = 1388$)	✗	✓($\eta = 0.0033$)	✓($\eta = 0.0125$)	✗
Fault II	✓($\eta = 0.0108$)	✗	✓($\eta = 1656$)	✗	✗	✓($\eta = 0.0084$)	✓($\eta = 0.0027$)
Fault III	✓($\eta = 0.0027$)	✓($\eta = 0.0054$)	✓($\eta = 0.8335$)	✗	✓($\eta = 0.0049$)	✓($\eta = 0.0273$)	✗

Table 6.4 Results of the 3-Factor ANOVAs for the monitoring performance evaluation. ✓indicates statistical significant differences ($p - value < 0.05$). ✗ is used when there are no statistical significant differences.

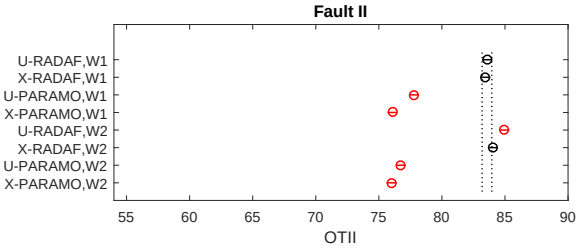
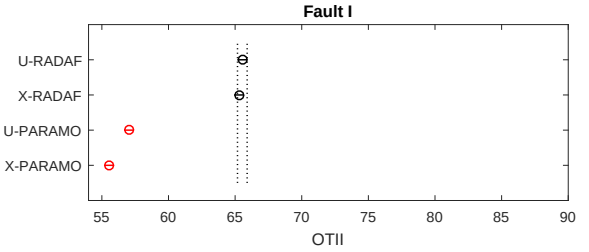
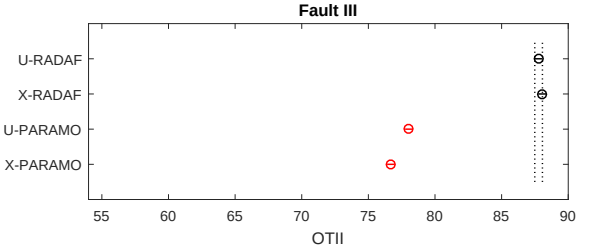


Fig. 6.12 LSD intervals at 95% confidence level of the OTII values computed for Fault II for the interaction between enhancing approach, moving window methods and the selected configuration settings.



(a)



(b)

Fig. 6.13 LSD intervals at 95% confidence level of the OTII values for the interaction between enhancing approach and moving window methods computed for (a) Fault I and (b) Fault III.

Does PARAMO outperform TCS from the point of view of the parameter stability?

An ANOVA study of one factor (the pre-processing method, f) at five levels (see Table 6.5) is performed individually for means, standard deviations and loadings. The ANOVA results show that there are statistical significant differences. The LSD intervals in Fig. 6.14 show that the NSDs obtained with PARAMO outperform TCS, and their differences are statistically significant ($p - value < 0.05$) for the means and standard deviations.

Factor	*Level*	*Description*
f	TCS U-PARAMO, W1 U-PARAMO, W2 X-PARAMO, W1 X-PARAMO, W2	Pre-processing method

Table 6.5 Levels for the 1-factor ANOVA studies.

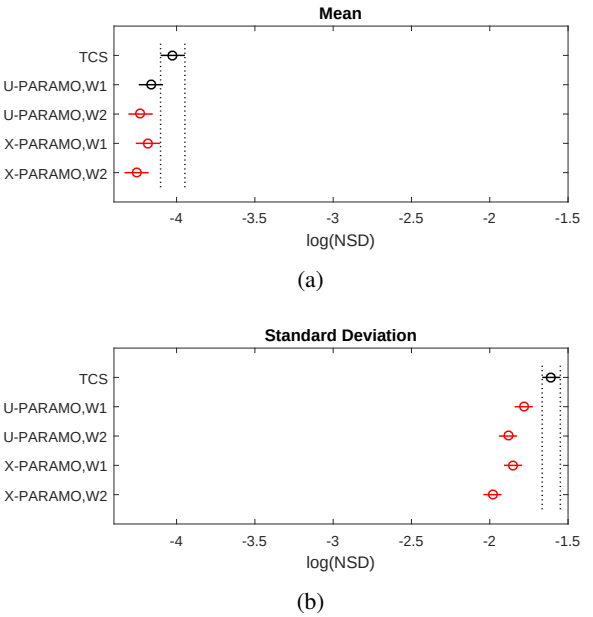
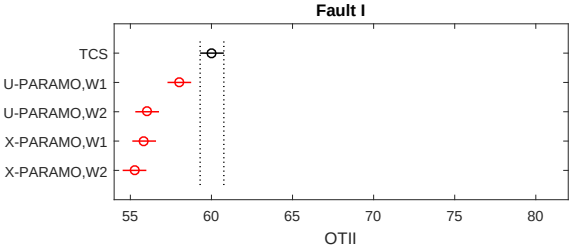


Fig. 6.14 LSD intervals at 95% confidence level of the NSD values computed for the simple effect of type of pre-processing (TCS and PARAMO) for (a) the means and (b) standard deviations.

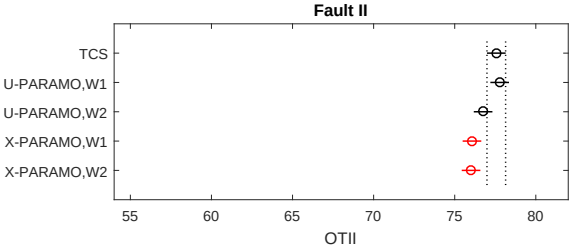
Does PARAMO outperform TCS from the point of view of monitoring?

An ANOVA of a single factor at five levels (see Table 6.5) is performed separated for each generated fault. The ANOVAs unveil that there exist statistical significant differences among methods for the faults under study.

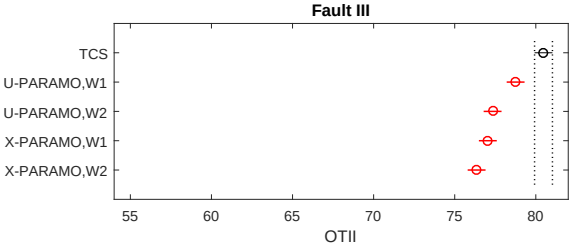
The resulting LSD values in Fig. 6.15 show that the OTII values obtained with X-PARAMO are consistently lower than for TCS, irrespective of the settings under study.



(a)



(b)



(c)

Fig. 6.15 LSD intervals at 95% confidence level of the OTII values for the simple effect of type of pre-processing (TCS and PARAMO) computed for (a) Fault I, (b) Fault II and (c) Fault III.

These outcomes support the claim that X-PARAMO not only reduces the parameter uncertainty, but also improves the accuracy of the monitoring systems to detect faults.

6.7.2 Results applying RADAF in model building

In this part of the section, we consider the use of RADAF also to obtain the loadings from filtered data. That is, unlike in the application of RADAF in the previous sub-section, we also use filtered data (Equations 6.20 and 6.23) to fit the PCA model. This causes a delay of $\lfloor W/2 \rfloor$ in the application of this approach in real time, due to test data also has to be filtered.

After the model building from the filtered data, the parameter stability is assessed again for the loadings (note that the results does not change for the means and standard deviations, as they are computed exactly like in Section 6.7.1). There are statistical significant differences between PARAMO and RADAF in the same factors and interactions as in Section 6.7.1. However, in this case, RADAF is better than PARAMO and setting $W2$ is better than $W1$. The LSDs are displayed in Fig. 6.16.

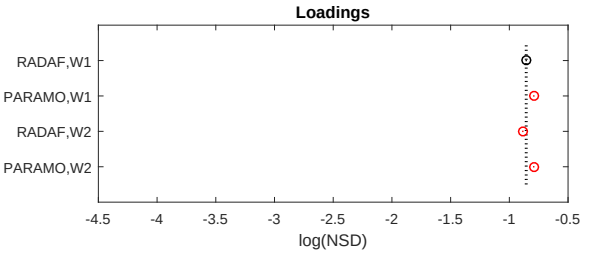


Fig. 6.16 LSD intervals at 95% confidence level of the NSD values for the interactions between enhancing approach and the selected configuration settings computed the loadings. RADAF is applied to obtain the pre-processing parameters and for to build the PCA model

In comparison with TCS, there are also statistical significant differences, being RADAF better than TCS. The corresponding LSDs are shown in Fig. 6.17.

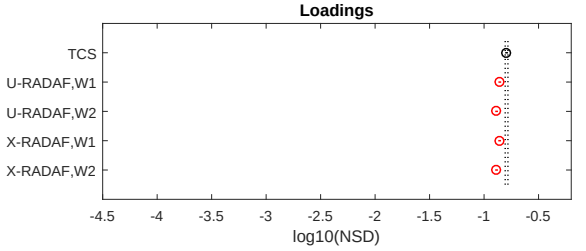
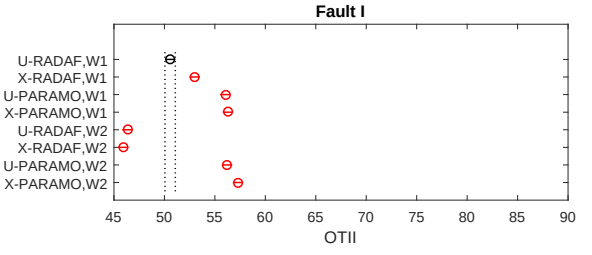


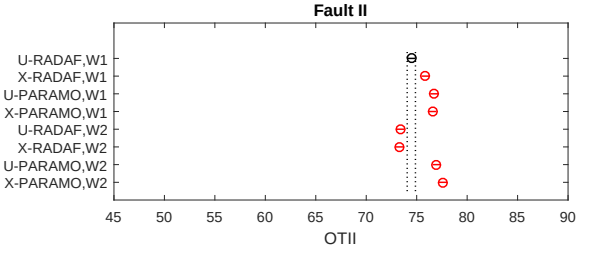
Fig. 6.17 LSD intervals at 95% confidence level of the NSD values for the simple effect of type of pre-processing (TCS, RADAF and PARAMO) computed for the loadings. RADAF is applied to obtain the pre-processing parameters and for to build the PCA model.

A 3-factor ANOVA with interactions is performed separated for each generated fault. The ANOVAs unveil that there exist interactions between all the factors under study and the interactions are significant. The resulting LSD values in Fig. 6.18 show that the OTII values obtained with RADAF are consistently lower than for PARAMO for Fault I and II, irrespective of the setting and the moving window method. For Fault III, the OTII values obtained with RADAF are consistently lower than for PARAMO for W2, while for W1, there are no statistically significant differences between U-RADAF and X-PARAMO and U-PARAMO is statistically significant better than X-RADAF.

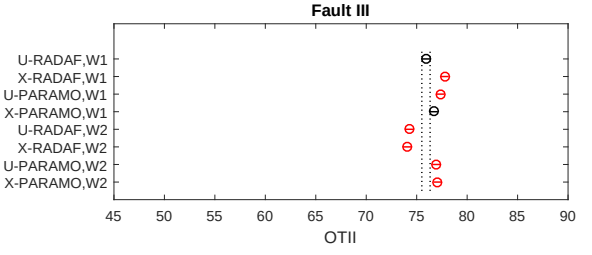
Finally, an ANOVA of a single factor at five levels is performed separated for each generated fault. The ANOVAs unveil that there exist statistical significant differences among methods for the faults under study. The resulting LSD values in Fig. 6.19 show that the OTII values obtained with RADAF are consistently lower than for TCS, irrespective of the settings or the moving



(a)



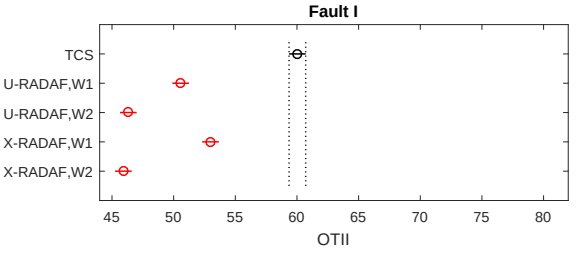
(b)



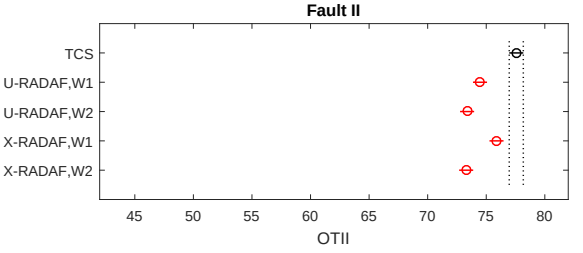
(c)

Fig. 6.18 LSD intervals at 95% confidence level of the OTII values for the interactions between enhancing approach PARAMO and RADAF, the moving window method (Uniform and eXponential), and the selected configuration settings W1 and W2 computed on (a) Fault I, (b) Fault II and (c) Fault III.

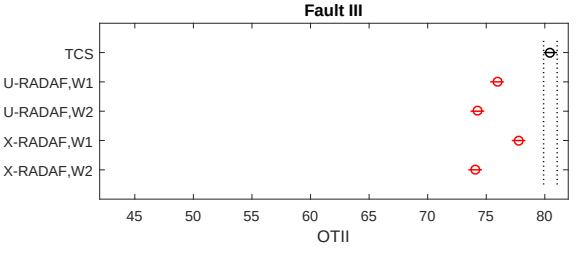
window method under study. The best results for RADAF correspond to setting W2.



(a)



(b)



(c)

Fig. 6.19 LSD intervals at 95% confidence level of the OTII values for the simple effect of type of pre-processing (TCS, PARAMO and RADAF) computed for (a) Fault I, (b) Fault II and (c) Fault III.

It can be concluded that filtering the data improves the loadings and so the model does, with the corresponding enhancing in the capability of fault detection. In our experiments, these results are better as the filtering is more aggressive.

6.8 Conclusions

This chapter addresses two open points related to the parameter stability and its impact on process modeling: *i)* the development of novel pre-processing approaches to enhance model parameter stability, and *ii)* the study of the influence of parameter stability on fault detection.

More precisely, the comparative study here unveils that the approach based on obtaining the pre-processing parameters from more observations, PARAMO, outperforms the established methodology for pre-processing batch data in Batch Multivariate Statistical Process Control. Using this proposal, both the parameter stability and the monitoring performance are enhanced. Thus, we show that it is possible to enhance the parameter stability by reducing the uncertainty in the pre-processing parameters; and that the lower the uncertainty in the model parameters, the higher the quality in the monitoring system.

The results improve even more when RADAF is applied also in model building accepting a $\lfloor W/2 \rfloor$ delay. This implies that, for the *Saccharomyces Cerevisiae* cultivation process, a window of $W = 3$ produces a delay of $\approx 10'$, and for $W = 5$ the delay is $\approx 20'$. These times may not be affordable for this type of process. The decision of applying this approach should be made in coherence with the needs of improving the pre-processing parameters, and always remembering that the delay increases also with the degree of the smoothing. Another solution to face up the symmetric filtering in online monitoring might be using missing data imputation [106]. On the contrary, PARAMO ensures an on-line monitoring without any delay.

Finally, we must remark that, despite this study has been carried out in the MSPC context, it is applicable in a straightforward manner to MSNM. Indeed, these pre-processing techniques are applied in Chapter 8 to real network data.

7

Diagnosis

"Life does not deserve one to worry so much."
Marie Curie, Nobel Prize in Physics in 1903

"Everything should be simplified as much as possible, but no more."
Albert Einstein, Nobel Prize in Physics in 1921.

This chapter is mainly based on the following research paper:

- **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2018). Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control. *Chemometrics and Intelligent Laboratory Systems*, 172:194–210

Contents

7.1 State-of-the-art Diagnosis Methods 147
7.1.1 Contribution Plots (CP) 147
7.1.2 Reconstruction-Based Contributions (RBC) . . . 148
7.1.3 The observation-based Missing-data method for Exploratory Data Analysis (oMEDA) 149
7.2 Univariate Squared: a Different Approach for Diagnosis 152
7.3 Methodology for Comparison of Diagnosis Methods . 153
7.3.1 Step 1. Generation of Anomalies with Known Diagnosis 154
7.3.2 Step 2. Definition of a metric to evaluate the diagnosis performance 156
7.3.3 Step 3. Experimental Design 157
7.4 Evaluation of Diagnosis Methods 159
7.4.1 Case of Study I: Simulated Synthetic Data 160
7.4.2 Case of Study II: Simulated Communication Network Traffic 164
7.4.3 Case of Study III: Data Set for Process Control . 169
7.5 Conclusions 177

One of the most important steps in a monitoring system is to identify the variables related to a previously detected anomaly. This step is that is

termed **diagnosis** of the anomaly, and it helps the analysts to identify the root causes of the anomaly so that problems within the process or the network can be timely identified and corrected for [117]. As already stated in Chapter 4, the diagnosis takes even a more important role in MSNM, since the number of security events is so high that security operators usually cannot handle all of them. This makes necessary the prioritization and triage of the events, which can be performed in a more efficient manner thanks to the diagnosis.

This chapter starts by reviewing the state-of-the-art diagnosis methods in the MSPC. Then, the two contributions of this part are presented. On the one hand, a method that follows a univariate approach to enhance the diagnosis process is introduced. On the other hand, a methodology for the comparison of diagnosis methods is proposed. This methodology is applied over three state-of-the-art multivariate diagnosis methods and the univariate proposal.

7.1 State-of-the-art Diagnosis Methods

There have been a number of developments in diagnosis in the past decades. Three existing multivariate methods for MSPC diagnosis are described in this section. On a general perspective, reviewed works provide limited comparison with other approaches in the literature. In addition, multivariate methods suffer from the *smearing* problem: misdiagnosis owing to the spread of the contribution of the variables affected by an anomaly to those not affected by it [114, 212]. This problem results in a more complex diagnosis process and reinforces the necessity of a comprehensive study that compares these techniques.

7.1.1 Contribution Plots (CP)

The CP is currently the most accepted approach for diagnosis in PCA-MSPC [117, 150, 212, 213]. The contribution of the m-th variable to the D-statistic, c_m^D, is

obtained from the following expression:

$$c_m^D = \mathbf{t} \cdot \Lambda^{-1} \cdot \mathbf{p}'_m \cdot x_m \tag{7.1}$$

where $\mathbf{p}_m$ is the vector in the m-th row of the *loading* matrix for the A selected PCs and x_m is the value of the m-th variable in an (anomalous) observation, $\mathbf{x}$.

The contribution of the m-th variable to the Q-statistic, c_m^Q, corresponding to the residual, is calculated applying:

$$c_m^Q = (x_m - \mathbf{p}_m \cdot \mathbf{t}')^2 \tag{7.2}$$

7.1.2 Reconstruction-Based Contributions (RBC)

The RBC is a popular method that follows an alternative approach to compute the contributions of the variables to a given statistic [4, 5]. It is based on the work of Dunia *et al.* [69]. The contribution of the m-th variable to the D-statistic, rbc_m^D, corresponding to the model, is calculated from the expression:

$$rbc_m^D = \frac{(\mathbf{i}_m \cdot \mathbf{D}_A \cdot \mathbf{x}')^2}{d_{mm}} \tag{7.3}$$

where $\mathbf{D}_A = \mathbf{P}_A \cdot \Lambda^{-1} \cdot \mathbf{P}'_A$, $\mathbf{i}_m$ stands for the m-th row vector of the identity matrix $\mathbf{I}$ with size $M \times M$, and d_{mm} is the m-th element in the main diagonal of matrix $\mathbf{D}_A$.

The contribution rbc_m^Q to the Q-statistic, corresponding to the residual, is obtained from:

$$rbc_m^Q = \frac{(\mathbf{i}_m \cdot \mathbf{C}_R \cdot \mathbf{x}')^2}{c_{mm}^R} \tag{7.4}$$

with $\mathbf{C}_R = \mathbf{P}_R \cdot \mathbf{P}'_R$, being $\mathbf{P}_R$ the *loading* matrix with the residual components from $A+1$ to M, and c_{mm}^R the m-th element in the main diagonal of matrix $\mathbf{C}_R$.

7.1.3 The observation-based Missing-data method for Exploratory Data Analysis (oMEDA)

The oMEDA is a more recent method that was originally designed for exploratory data analysis. This method computes the contribution of a variable to specific patterns, such as clusters or outliers, in the scores distribution [29]. Unlike the previous methods, it uses the same expression for the model and residual sub-spaces and it does not compute the contributions to the statistics. Moreover, it handles groups of observations if desired.

Let us consider the column vector, $\mathbf{x}_m$, containing the values for the m-th variable in the group of observations to be diagnosed:

$$\mathbf{x}_m = \hat{\mathbf{x}}_m(Z) + \mathbf{e}_m(Z) \tag{7.5}$$

where $\hat{\mathbf{x}}_m(Z)$ is the projection of $\mathbf{x}_m$ in a given sub-space Z and $\mathbf{e}_m(Z)$ is the corresponding residual. Then, oMEDA follows:

$$d_m^Z = 2 \cdot \mathbf{x}'_m \cdot \mathbf{D} \cdot |\hat{\mathbf{x}}_m(Z)| - \hat{\mathbf{x}}'_m \cdot \mathbf{D} \cdot |\hat{\mathbf{x}}_m(Z)| \tag{7.6}$$

that can be re-expressed only in terms of $\mathbf{x}_m$ and $\mathbf{e}_m(Z)$, from Equation (7.5) as follows:

$$d_m^Z = (\mathbf{x}'_m + \mathbf{e}'_m(Z)) \cdot \mathbf{D} \cdot |\mathbf{x}_m - \mathbf{e}_m(Z)| \tag{7.7}$$

where $\mathbf{D} = \frac{\mathbf{d} \cdot \mathbf{d}'}{\|\mathbf{d}\|^2}$ and $\mathbf{d}$ is a dummy column vector with non-zero values in positions corresponding to the observations to be studied [1].

Fig. 7.1 shows the result of applying oMEDA following the same example as in the original work (archaeological artifacts dataset) [29]. In this dataset, the concentration of ten different metals in archaeological artifacts is provided with the objective of identifying whether there exist any differences and/or

[1] The way of selecting the possible non-zero values is out of the scope of this work. For further details refer to the original paper [29].

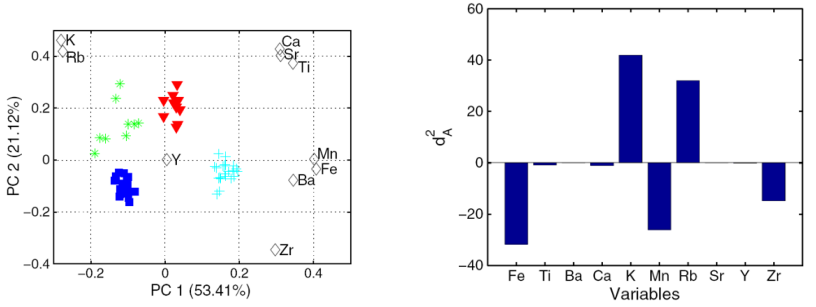
similarities among artifacts in different quarries or not. The artifacts are 63 obsidian samples obtained from four known quarries. First, a PCA is performed. Fig. 7.1 (a) shows the two first components of the model, capturing the 70% of the variance approximately. In this figure, we can observe that metals are grouped (*e.g.* `Ca`, `Sr` and `Ti` or `Mn`, `Fe` and `Ba`). Furthermore, we can distinguish the distinct quarries depicted with different colors and forms: first quarry is represented by red inverted triangles, the second one by green stars, the third one by blue squares, and the fourth one by cyan pluses. After obtaining the PCA model, we apply oMEDA to investigate the relationships among observations and variables. To do this, we use again the first two components of the model and, as an example, we compare first and fourth quarries. Then, we assign zero to all the observations that we are not interested in (second and third quarries), and different values to the groups of variables that we want to compare (first quarry = 1, fourth quarry = -1). Fig. 7.1 (b) shows the result of applying oMEDA in that way. We can observe that samples from the first quarry have higher amounts of `K` and `Rb` than the fourth one (positive bars). On the contrary, `Fe`, `Mn` and `Zr` have higher importance in the fourth quarry than in the first one (negative bars).

oMEDA can be used for the diagnosis of one observation by filling $\mathbf{D}$ with zeros in all the observations except for that of interest to diagnose. This is equivalent to define $\mathbf{d} = 1$, which in turn means, $\mathbf{D} = 1$, if oMEDA is applied only to the observation of interest. Then, the expressions corresponding to the two sub-spaces under study can be obtained by substituting $\mathbf{D} = 1$ in Equation (7.7):

$$d_m^{\mathbb{D}} = (x_m + e_m) \cdot |x_m - e_m| \tag{7.8}$$

where $Z = \mathbb{D}$ refers to the model sub-space,

$$d_m^{\mathbb{Q}} = (x_m + \hat{x}_m) \cdot |x_m - \hat{x}_m| \tag{7.9}$$



(a) (b)

Fig. 7.1 Example of grouping observations applying oMEDA, showing (a) Bi-plot for the first two PCs of the PCA of the archaeological dataset [29] and (b) Comparison of the first two PCs for two observations.

and $Z = \mathbb{Q}$, refers to the residual sub-space[2].

Let us consider again the example in Chapter 3 (see Fig. 7.2). In that example, first we generated randomly a 100×10 calibration matrix. Then, we also created a 10×10 test matrix that follows the same correlation pattern as in the calibration. The value of the first two variables in the last five test observations is increased to make them anomalous. We only diagnose the last one. To do this, we create a dummy vector with all the observations having zero value, excepting the one we want to diagnose: $\mathbf{d} = [0, 0, ..., 0, 0, 1]$. This allows us to identify the anomalous variables (see Section 3.2). Those variables showing a high magnitude (in absolute value) are related to the anomaly. Fig. 7.2 shows two bars corresponding to the first two variables of the test data, which are the ones that we previously turned anomalous.

[2]Note that the superscripts $\mathbb{D}$ and $\mathbb{Q}$ are used to maintain the consistency with the terminology used in the previously studied diagnosis methods.

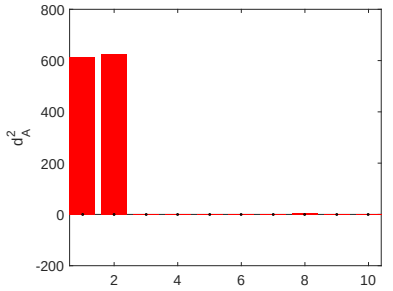


Fig. 7.2 Example of diagnosis applying oMEDA.

7.2 Univariate Squared: a Different Approach for Diagnosis

After an anomaly occurs, the correlation structure in the model may not hold for such anomaly and therefore the division in model/residuals found for calibration data may not be optimum for diagnosis. If this occurs, one can consider that it makes no sense to calculate the contribution of the variables to each statistic separately [195].

Under this hypothesis, it might be interesting to take into account the full variable space for diagnosis. The fact that oMEDA is equally computed for the model and residual sub-spaces makes its extension to the complete variable space straightforward. Thus, setting $Z = \mathbb{D} + \mathbb{Q}$, and using Equations (7.8) and (7.9), the following expression yields:

$$u_m = d_m^{\mathbb{D}+\mathbb{Q}} = x_m \cdot |x_m| \tag{7.10}$$

being equivalent to $u_m = sign(x_m) \cdot (x_m)^2$, which we call *Univariate-Squared*.

Note that this expression corresponds to a univariate approach because it considers only the original value of each variable and not the scores. Similar

approaches are analyzed elsewhere [113, 114] but their performance has not been assessed properly through a thorough comparison.

The univariate proposal contrasts with the accepted trend in PCA-based MSPC diagnosis: it adopts a univariate approach although a multivariate detection has been previously applied. The main advantage of this method is that it does not suffer from the smearing problem, as the correlation, which is the main cause of the smearing [113, 114, 212], is not considered in the computation.

7.3 Methodology for Comparison of Diagnosis Methods

While there have been a number of developments in diagnosis in the past decades [4, 5, 29, 113, 114, 117, 122, 123, 150, 155, 166, 213], no sound method for comparing existing approaches has been proposed.

In this section, a procedure for comparison of diagnosis methods is presented. This methodology is inspired in the field of Design of Experiments and Analysis of Variance [142] and enables consideration of the different factors that might influence the performance of diagnosis methods, providing a comparison framework. This method has been developed to meet the requirements that, in our opinion, a comprehensive comparison methodology should meet:

- *Generation of anomalies with known diagnosis*. The variables related to an anomaly should be known in advance, so that it is possible to check whether the diagnosis methods identify these variables correctly or not. This provides a ground truth of such affected variables.
- *Definition of a metric to evaluate the diagnosis performance*. A quantitative measure to assess the performance of diagnosis method is needed.

- *Experimental Design*. The factors that might have an impact on the methods under consideration should be assessed properly using an adequate design. Additionally, it is needed to define a number of replicates of the experiment, to *reduce the uncertainty of the results*.

7.3.1 Step 1. Generation of Anomalies with Known Diagnosis

We propose a procedure to modify NOC observations to obtain anomalies. In principle, the use of NOC observations as starting point ensures that there are no other anomalies in the observation except those introduced artificially. The procedure for the artificial generation of anomalies is described in the next paragraphs.

Let us consider an anomaly-free observation from the NOC matrix, $\mathbf{x} \in \mathbf{X}_{NOC}$, and the set of variables to be altered, $\tilde{\mathbf{V}}$. The observations are split in the columns into two parts. The sub-vector corresponding to $\tilde{\mathbf{V}}$ (the variables to be altered) is denoted by $\tilde{\mathbf{x}}$, while the sub-vector including those variables not modified is $\dot{\mathbf{x}}$. Then, the anomalous observation, $\mathbf{x}_{alt}$, is obtained by altering the original value of $\mathbf{x}$ as follows:

$$\mathbf{x}_{alt} = \mathbf{x} + \mathbf{r} \tag{7.11}$$

where the value of each variable m is computed following:

$$x_{alt,m} = \begin{cases} x_m, & \text{if } x_m \in \dot{\mathbf{x}}, \\ \chi \cdot s, & \text{if } x_m \in \tilde{\mathbf{x}}, \end{cases} \tag{7.12}$$

with $s = sign(x_m)$, the sign of the variable m in the original observation. χ is the altered value of the observation, and in this work it is the same for each variable $m \in \tilde{\mathbf{V}}$. This assumption is not perfect, but a Monte Carlo approach as described in Section 7.3.3 provides enough variability to obtain realistic and significant results. The Equation (7.12) is applied to modify the original value

of $\mathbf{x}$, using a value χ so that one of the statistics is equal to its corresponding UCL multiplied by a given factor, K. This allows to obtain an anomalous observation, $\mathbf{x}_{alt}$ following the definition of MSPC. This can be done in (at least) two ways:

- *Trial and error*. χ is iteratively increased to modify the normal value of the selected variables until any of the statistics is equal to K·UCL.
- *Analytically*. χ is computed applying analytic expressions to alter the selected variables.

Since the numeric approach can be computationally intensive, the analytic alternative is generally preferred.

To derive the analytical expression, let us start by analyzing the equation applied to compute the D-statistic, $Dst = \mathbf{t} \cdot \Lambda^{-1} \cdot \mathbf{t}'$, where $\mathbf{t}$ is the score for the observation, $\mathbf{x}$, to be altered. Considering that $\mathbf{t} = \mathbf{x} \cdot \mathbf{P}_A$, the vector can be re-ordered into affected and non-affected variables: $\mathbf{x}_{alt} = [\dot{\mathbf{x}}\ \tilde{\mathbf{x}}]$ and the corresponding re-ordered *loading* matrix follows: $\mathbf{P}_A = \begin{bmatrix} \dot{\mathbf{P}}_A \\ \tilde{\mathbf{P}}_A \end{bmatrix}$, where $\tilde{\mathbf{P}}_A$ and $\dot{\mathbf{P}}_A$ are the loadings for the altered and non-altered variables, respectively. A re-defined expression for the D-statistic is:

$$Dst = (\dot{\mathbf{x}} \cdot \dot{\mathbf{P}}_A + \tilde{\mathbf{x}} \cdot \tilde{\mathbf{P}}_A) \cdot \Lambda^{-1} \cdot (\dot{\mathbf{P}}'_A \cdot \dot{\mathbf{x}}' + \tilde{\mathbf{P}}'_A \cdot \tilde{\mathbf{x}}') \tag{7.13}$$

For fixed $\dot{\mathbf{x}}$, solving the equation for $Dst = K \cdot UCL_D$ provides the value for $\tilde{\mathbf{x}}$ from the quadratic expression:

$$Dst = d^D + \tilde{\mathbf{x}} \cdot b^D + \tilde{\mathbf{x}}^2 \cdot a^D = K \cdot UCL_D \tag{7.14}$$

with $d^D = \dot{\mathbf{x}} \cdot \dot{\mathbf{P}}_A \cdot \Lambda^{-1} \cdot \dot{\mathbf{P}}'_A \cdot \dot{\mathbf{x}}'$, $b^D = \mathbf{s} \cdot (2 \cdot \tilde{\mathbf{P}}_A \cdot \Lambda^{-1} \cdot \dot{\mathbf{P}}'_A \cdot \dot{\mathbf{x}}')$, $a^D = \mathbf{s} \cdot \tilde{\mathbf{P}}_A \cdot \Lambda^{-1} \cdot \tilde{\mathbf{P}}'_A \cdot \mathbf{s}'$, and $\mathbf{s}$ the vector containing the original sign of each variable in $\mathbf{x}$. Finally, the value for the observation $\mathbf{x}$ after applying the alteration, $\mathbf{x}_{alt}$, is

obtained by replacing the variables to be modified with the result of solving Equation (7.14) as a normal quadratic equation and selecting the solution that keeps the positive sign in the discriminant:

$$\tilde{x}^D = (-b^D + \sqrt{(b^D)^2 - 4 \cdot a^D \cdot c^D})/(2 \cdot a^D) \tag{7.15}$$

where $\tilde{x}^D$ is the new value assigned to each selected variable for the altered observation, $\mathbf{x}_{alt}$, and $c^D = d^D - K \cdot UCL_D$.

Similarly, to alter a given observation, $\mathbf{x}$, for the Q-statistic it is necessary to replace the equation to solve, $Qst = K \cdot UCL_Q$ and to consider $Qst = \mathbf{t}_R \cdot \mathbf{t}'_R$. This makes $d^Q = \dot{\mathbf{x}} \cdot \dot{\mathbf{P}}_R \cdot \dot{\mathbf{P}}'_R \cdot \dot{\mathbf{x}}'$, $c^Q = d^Q - K \cdot UCL_Q$, $b^Q = \mathbf{s} \cdot (2 \cdot \tilde{\mathbf{P}}_R \cdot \dot{\mathbf{P}}'_R \cdot \dot{\mathbf{x}}')$ and $a^Q = \mathbf{s} \cdot \tilde{\mathbf{P}}_R \cdot \tilde{\mathbf{P}}'_R \cdot \mathbf{s}'$, where $\mathbf{t}_R$ and $\mathbf{P}_R$ stand for the scores and the loadings in the residual components. The resulting expression is:

$$\tilde{x}^Q = (-b^Q + \sqrt{(b^Q)^2 - 4 \cdot a^Q \cdot c^Q})/(2 \cdot a^Q) \tag{7.16}$$

Finally, the new value for the variables to be altered is:

$$\chi = \min(\tilde{x}^D, \tilde{x}^Q) \tag{7.17}$$

7.3.2 Step 2. Definition of a metric to evaluate the diagnosis performance

In order to know if a diagnosis method outperforms another, it is needed to quantify their performance in a meaningful way. In the present work, a ratio is defined, based on the relation between the contribution of the anomalous variables and the contribution of the non-affected variables. The proposed metric is calculated as the quotient between the average of the contributions from these variables, which is denoted *Diagnosis Goodness Ratio*, γ:

$$\gamma = \frac{\mu_{\tilde{\mathbf{c}}}}{\mu_{\dot{\mathbf{c}}}} \tag{7.18}$$

with $\mu_{\tilde{\mathbf{c}}}$ and $\mu_{\dot{\mathbf{c}}}$ computed as follows:

$$\mu_{\tilde{\mathbf{c}}} = \frac{\sum_{\tilde{x}_m \in \tilde{\mathbf{x}}} |c(\tilde{x}_m)|}{\mathscr{V}} \tag{7.19}$$

$$\mu_{\dot{\mathbf{c}}} = \frac{\sum_{\dot{x}_m \in \dot{\mathbf{x}}} |c(\dot{x}_m)|}{M - \mathscr{V}} \tag{7.20}$$

where $c(\tilde{x}_m)$ are the contributions for the affected variables, $c(\dot{x}_m)$ the contributions for the non-affected variables in the altered observation, $\mathbf{x}_{alt} = [\dot{\mathbf{x}}\ \tilde{\mathbf{x}}]$, and $\mathscr{V}$ is the number of altered variables. The greater the ratio γ is, the better the diagnosis performance of the method. On the contrary, if the value of γ is close or even lower than 1, there is no diagnosis capability at all.

7.3.3 Step 3. Experimental Design

Many elements can impact the diagnosis. The goal of this step is to evaluate how such factors affect the diagnosis performance. In this work, the following factors are considered:

- *Number of selected PCs* (*pcs*). The way of selecting the number of PCs to build a PCA-model still remains an open problem because the number of latent variables affects the quality of the model, the detection and the diagnosis performance [34, 35, 150, 213].
- *Number of variables to alter* ($\mathscr{V}$). The number of variables affected by an anomaly is expected to impact the diagnosis performance. Note that the expressions proposed in Equations (7.15) and (7.16) allow the alteration of any number of variables.
- The *relationship* between the *number of variables* (M) and the *number of samples* (N) in the calibration matrix, τ. For the present study, different types of matrices are considered: *Fat* matrices, F, with $M > N$; *Square* matrices, S, with $N \simeq M$; and *Thin* matrices, T, with $N > M$.

Considering the previous discussion, the Algorithm 1 is designed to compare a given set of diagnosis methods. When an experimental run is performed,

Algorithm 1 Core algorithm

procedure
  for each $\tau \in \{T, S, F\}$ **do**
    for each $pc \in pcs$ **do**
      for $v \in \{1 : \mathscr{V}\}$ **do**
        for each observation $n \in \{1 : N\}$ **do**
          $\mathbf{x} \leftarrow \mathbf{X}_{NOC}(n)$
          $\mathbf{x}_{alt} \leftarrow \mathbf{x}$ // Anomalous observation is initialized to $\mathbf{x}$
          $\tilde{\mathbf{V}} \leftarrow \{\tilde{v}_1, ..., \tilde{v}_v\}$ // Select $\tilde{v}_v$ randomly
          $\tilde{x}^D_{\tilde{\mathbf{V}}} \leftarrow$ Anomaly generation Eq.(7.15)
          $\tilde{x}^Q_{\tilde{\mathbf{V}}} \leftarrow$ Anomaly generation Eq.(7.16)
          $\chi \leftarrow \min\{\tilde{x}^D_{\tilde{\mathbf{V}}}, \tilde{x}^Q_{\tilde{\mathbf{V}}}\}$ Eq.(7.17)
          $\mathbf{x}_{alt}(\tilde{V}) \leftarrow \chi$ // Variables in $\tilde{V}$ take the new value, χ
          Compute $Dst(\mathbf{x}_{alt})$ and $Qst(\mathbf{x}_{alt})$
          Increase $nDst$ if $Dst > UCL_D$
          Increase $nQst$ if $Qst > UCL_Q$
          for each $method$ **do**
            Compute contributions
            $\gamma \leftarrow \frac{\mu_{\tilde{\mathfrak{c}}}}{\mu_{\dot{\mathfrak{c}}}}$ // Ratio calculation
          end for
        end for
      end for
    end for
  end for
end procedure

one way of increasing the confidence on the results is to repeat such experimental run a high number of times. The higher the number of repetitions, the more reliable the results. For this reason, a Monte Carlo procedure is applied over the core Algorithm 1, to perform an experimental comparison according to the identified needs, achieving low uncertainty results.

The Algorithm 1 is repeated over $\mathscr{Y}$ experimental runs by considering each combination of the parameters: the type of matrix (τ), the number of selected PCs (pcs), the number of variables to be altered ($\mathscr{V}$), and the randomly generated NOC observations (N). For each observation $\mathbf{x}$, v random variables are selected to obtain the set of variables $\tilde{\mathbf{V}}$ to be altered, where v varies in the range $\{1 : \mathscr{V}\}$ and corresponds to the number of selected variables. Once the anomaly is generated, it is introduced in these selected variables, producing $\mathbf{x}_{alt}$. Then, the statistics for the anomalous observation are computed, and we register those that exceed their corresponding UCL ($nDst$ and $nQst$) to compute the relative number of faults signaled by each of the statistics, the contributions and the ratio.

7.4 Evaluation of Diagnosis Methods

To assess the performance of the selected methods, the corresponding ratios are computed and compared under a wide range of simulated situations using simuleMV [31]. The results are validated using two real datasets: traffic data from a communications network [129], and another one obtained by simulating the *Saccharomyces cerevisiae* cultivation process [120] (see Chapter 5). The data are auto-scaled in all cases, since they include variables with incomparable units.

Note that the Monte Carlo approach allows the generation of anomalies that cover a wide range of possibilities, both univariate and multivariate and both maintaining/breaking the correlation structure in the model. Unlike in other related works [56, 57, 165, 166], the use of first principles models in the anomaly generation procedure is skipped to avoid drawing conclusions that only hold in very specific cases/processes. However, the results should be interpreted considering that there is no theoretical warranty that all types of failure are covered.

7.4.1 Case of Study I: Simulated Synthetic Data

The simulation software, simuleMV [31], generates random data for a given level of correlation, $L \in \{0,9\}$, where 0 means no correlation and 9 means the correlation is the maximum (recall Chapter 5). This software implements an algorithm that takes into account the number of observations, N, and the number of variables, M, for the matrix to be simulated. In addition, simuleMV enables the generation of a data matrix based on a given covariance matrix.

Table 7.1 shows the configuration for the experimental design using the proposed methodology.

τ	N	M	L	$\mathscr{Y}$	pcs	$\mathscr{V}$
Thin (T)	100	10	$\{3,6,9\}$	10	$\{1,2\}$	$\{1,2,3\}$
Square (S)	100	100	$\{3,6,9\}$	10	$\{1,4\}$	$\{1,2,3\}$
Fat (F)	100	1000	$\{3,6,9\}$	10	$\{1,11\}$	$\{1,2,3\}$

Table 7.1 Parameters involved in the Monte Carlo Simulation - L, N and M are parameters in simuleMV.

- Three types of matrices are simulated: T $(Thin) = 100 \times 10$, S $(Square) = 100 \times 100$, F $(Fat) = 100 \times 1000$.
- Three different correlation levels, L, are considered for each type of matrix: $low = 3$, $normal = 6$ and $high = 9$.
- $\mathscr{Y} = 10$ different replicates are generated for each type of matrix and correlation level.
- The number of selected PCs is: *i)* $pcs = 1$, and *ii)* the number of PCs that captures the 75% of the total variance in most of the replicates.
- The number of variables to be altered, $\mathscr{V}$, is varied from 1 to 3.

The performance of the diagnosis of statistics Q and D is assessed separately, as customary in the literature. Following the expressions defined in

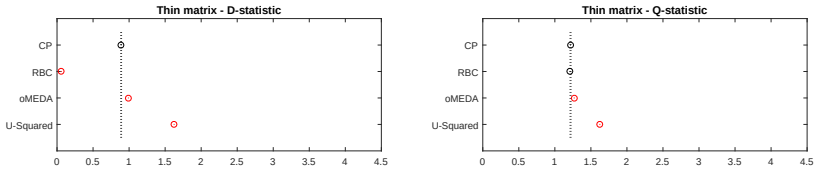
Equations (7.15) and (7.16), the variables are altered until any of the statistics is $K = 2$ times its upper control limit. Algorithm 1 is applied iteratively over the presented parameters. Although we focus in making anomalous one of the statistics as described above, the anomaly may be detected by both statistics at the same time. The diagnosis is applied only for those statistics in which the anomaly was manifesting.

Results

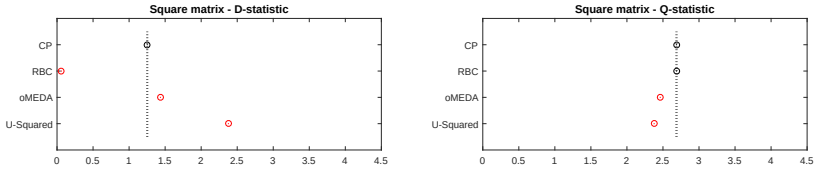
The comparison study includes an ANOVA performed on the ratio values. A logarithmic transform is applied to the ratio outcomes to smooth their positive skewness. The test includes the main factors of the experiment and first-order interactions: correlation level (L), selected PCs (pcs), number of affected variables ($\mathscr{V}$), diagnosis method, type of matrix (τ), and statistics. The ANOVA results show that all these factors and their corresponding interactions, except the correlation level, are statistically significant ($p - value < 0.01$).

It is also interesting to identify which of the studied parameters are most relevant. With this aim, the effect size is computed as in the previous chapter ($\eta^2 = SS(f)/SS(total)$). The most relevant parameters, sorted by η^2, are the type of matrix (τ), the statistic, and the diagnosis method. These parameters also present strong interactions; thus, varying any of them has a considerable effect on the other. This suggests that the comparison of the diagnosis methods should be performed individually for each combination of statistic and type of matrix. The LSD plots for statistical significant differences are computed and shown in Fig. 7.3. *U-Squared* in most cases outperforms the other methods, except for Square matrices and the Q-statistic, where CP and RBC present better results.

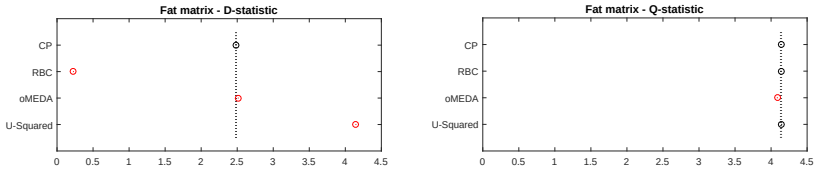
Results should be interpreted taking into account whether the anomalies are generated in the D-statistic, the Q-statistic or both. The percentage of generation for a normal correlation level, $L = 6$, is shown in Fig. 7.4. In



(a) Thin matrix (100x10)



(b) Square matrix (100x100)

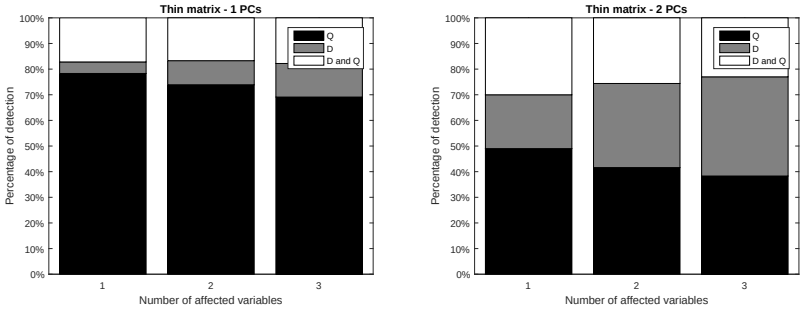


(c) Fat matrix (100x1000)

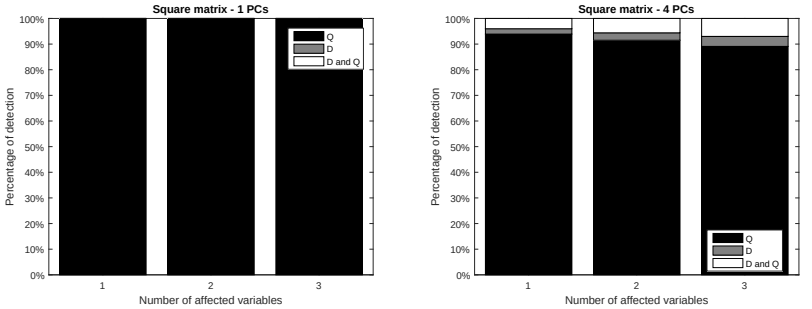
Fig. 7.3 ANOVA indicates that the results are significant for the selected parameters. LSD plots for (a) Thin matrices (T), (b) Square matrices (S), and (c) Fat matrices (F). The results for the D-statistic are shown in the left column, while the right column displays those results for the Q-statistic.

general, the probability of generating an anomaly in the D-statistic increases with the number of affected variables and the number of selected PCs whereas the percentage of generating an anomaly in the Q-statistic is always higher than that of the D-statistic in our experiments. Though not shown in the figure, this trend is observed to grow when the correlation level is increased.

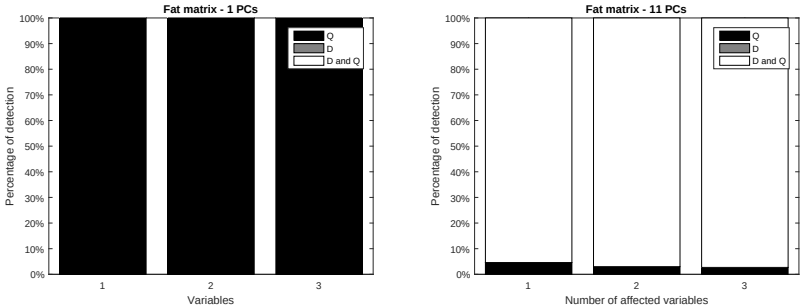
Since statistical significance is not the same as practical significance [142]. Fig. 7.5 and 7.6 show the results for a normal correlation level, $L = 6$, in the form of Box plots. The aforementioned plots are produced using the Advance Box Plot library, *aboxplot* [21]. These plots include the mean value



(a) Thin matrices (100x10)



(b) Square matrices (100x100)



(c) Fat matrices (100x1000)

Fig. 7.4 Percentage of anomalies generated on each statistic for 1 PC and for the number of PCs that captures 75% of the total variance: 2 PCs, 4 PCs, and 11 PCs for (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) simulated with correlation level $L = 6$ (normal correlation) in SimuleMV.

represented by a circle, together with the quartiles and outliers. The ratios, γ, are displayed for each type of matrix, statistic and number of PCs. Since there are only a few or no anomalies in the D-statistic when 1 PC is selected for the Square and Fat matrices, see Fig. 7.4, those results are not shown in Fig. 7.5, and only the ratios for the Q-statistic for Square and Fat matrices are shown.

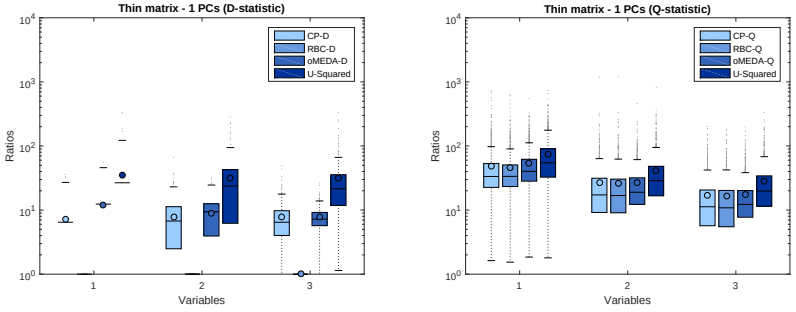
In general terms, the *Univariate-Squared* method provides results that are comparable with the rest of the methods in terms of the diagnosis ratio. U-Squared outperforms the other diagnosis methods for the D-statistic and, for Thin matrices, also for the Q-statistic. The differences between the results are more evident when the number of PCs is increased, which is also the reason for the larger differences in the D-statistic. The enhancement in the U-Squared ratio probably is due to the fact that correlation between the variables is not taken into account, and thus the *smearing* effect disappears.

Note that the *Reconstruction-Based Contributions* method has a very low ratio when the diagnosis is performed for the D-statistic. More precisely, when 1 PC is selected, the ratio γ is always equal to 1, indicating a complete lack of diagnosis capability. This result is mathematically proven by deriving the RBC expression for the D-statistic in the Appendix C [195].

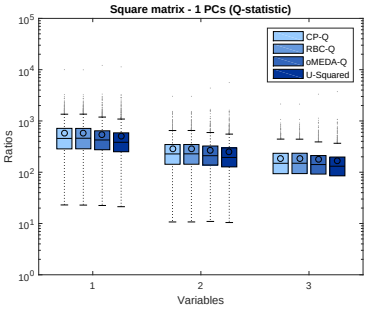
Finally, these results are repeated by using mean-centered data instead of auto-scaled (AS) data (not shown). Although the ratios are generally lower than for AS, the performance of the methods is the same as when using auto-scaled data.

7.4.2 Case of Study II: Simulated Communication Network Traffic

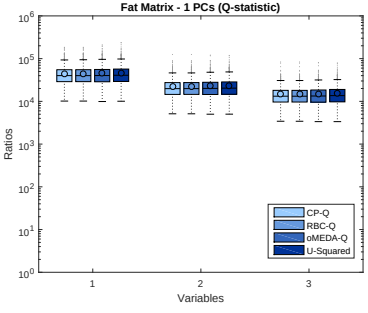
After performing the comparison with simulation data, the methodology is validated with data obtained from real applications to assess its consistency. More specifically, traffic data from a communications network [129] is considered.



(a) Thin matrices (100x10) - 1 PC

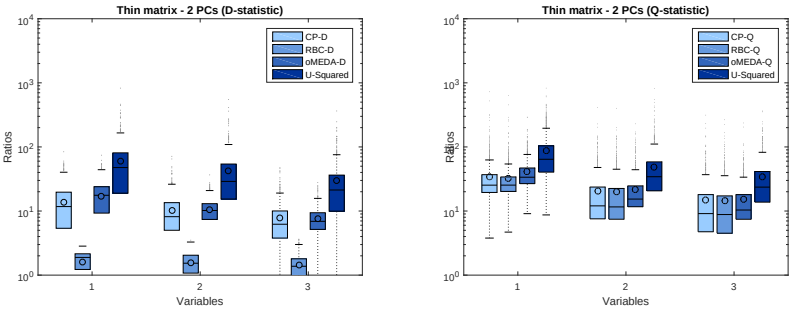


(b) Square matrices (100x100) - 1 PC

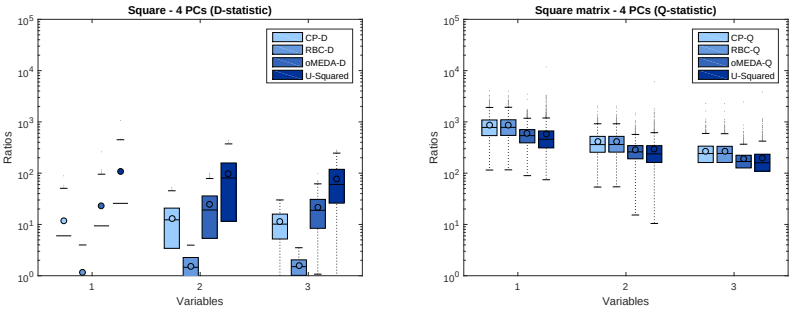


(c) Fat matrices (100x1000) - 1 PC

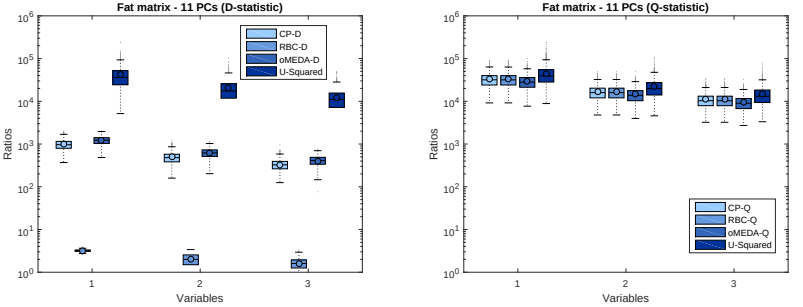
Fig. 7.5 Ratios, γ, for 1 PC and (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) simulated with correlation level $L = 6$ (normal correlation) in SimuleMV.



(a) Thin (100x10) - 2PCs



(b) Square (100x100) - 4 PCs



(c) Fat (100x1000) - 11 PCs

Fig. 7.6 Ratios, γ, for (a) 2 PCs and T, (b) 4 PCs and S, and (c) 11 PCs and F simulated with correlation level $L = 6$ (normal correlation) in SimuleMV.

This dataset is split into two sub-sets: one for the calibration, corresponding to one-third of the observations, $\mathbf{X}$, and another one with the remaining observations for testing, **test**. The matrix $\mathbf{X}$ contains $N = 501$ observations and $M = 24$ variables. The matrix **test** includes one hour with network attacks. Since our comparison approach uses NOC data, in order to avoid polluted values, the observations corresponding to attacks are removed. Then, only data below 50% of the UCL are used to ensure the test data are initially NOC. The final dataset, $\mathbf{X}_{NOC}$, has $N = 303$ observations and $M = 24$ variables.

The comparison algorithm (see Algorithm 1) is run using $N = 1000$ random observations. The number of selected PCs is *i)* $pcs = 1$, and *ii)* $pcs = 2$, which is the number of PCs that captures the 75% of the total variance. The variables are altered until either of statistics is $K = 2$ times its control limit. The configuration for the experiment is shown in Table 7.2.

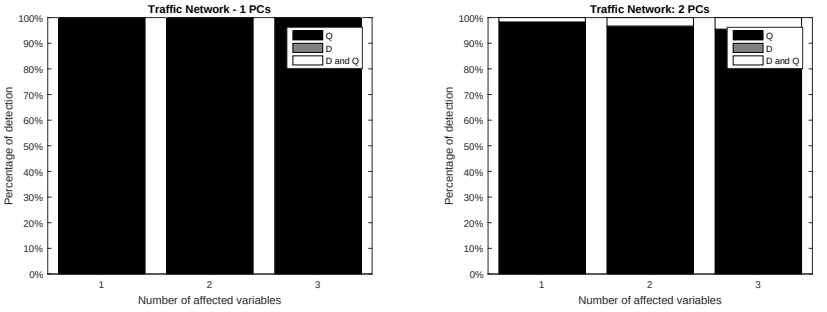
τ	N	M	$\mathscr{Y}$	pcs	$\mathscr{V}$
Thin (T)	303	24	1	$\{1,2\}$	$\{1,2,3\}$

Table 7.2 Parameters involved in the verification using *Network data*.

Results

Fig. 7.7 shows the percentages of anomaly generation for each statistic. The probability of generating of an anomaly only in the Q-statistic is closer to that of a Square matrix in the simulation result.

The ANOVA is performed on the ratio values to contrast the results with those from the simulated data. The test takes into account the factors of the experiment: selected PCs (pcs), number of affected variables ($\mathscr{V}$), diagnosis method, and statistics, as well as their first-order interactions. Note that the type of matrix is Thin in this case. The result from the test is consistent with that obtained using simuleMV and shows that all these factors and the corresponding interactions are statistically significant ($p - value < 0.01$).

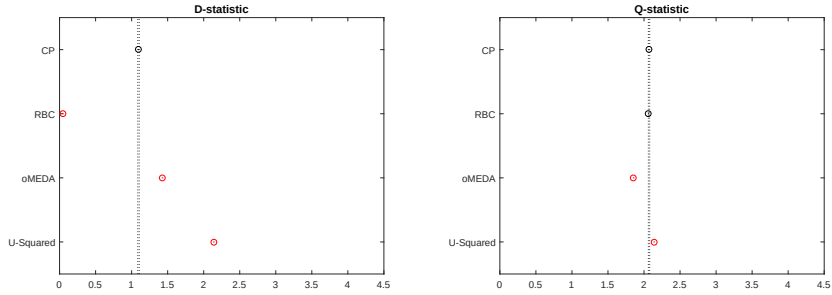


(a) (b)

Fig. 7.7 Percentage of anomalies generated on each statistic for (a) 1 PC and (b) 2 PCs corresponding to the *Communications Network Traffic* data set.

The most relevant parameters are, sorted by η^2, the statistic, the diagnosis method, and the number of altered variables. These parameters are, without considering the type of matrix, the same as those from the simulation and also present strong interactions. Using these results, the comparison is performed individually for the diagnosis methods for each statistic. When statistically significant differences are identified among the approaches, the LSD plots are computed (see Fig. 7.8). The results are congruent with those from Thin matrices in the simulation: U-Squared is better than the rest of the diagnosis methods.

Fig. 7.9 shows the distribution of the ratios computed after applying the diagnosis methods. According to the observed anomaly generation percentages for each statistic, there are only few or no anomalies in the D-statistic. Therefore, those are not considered, and only the ratios for the Q-statistic are shown in Fig. 7.9. U-Squared improves the multivariate diagnosis methods as the number of PCs and anomalous variables increase. The differences are more remarkable between U-Squared and oMEDA.



(a) (b)

Fig. 7.8 LSD plots for Network Traffic Data. The results for the D-statistic are shown in the left column, while the right column displays those results for the Q-statistic.

7.4.3 Case of Study III: Data Set for Process Control

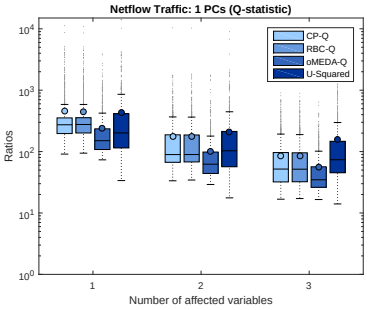
Similarly to the experiments carried out in Chapter 6, to validate the proposed methodology in MSPC, this study is carried out using process data. We use the data from the *Saccharomyces cerevisiae* batch process simulation [44, 120].

As the data are three-way, they are unfolded for the application of PCA-MSPC. *Batch-wise*, *Variable-wise* and *Batch-Dynamic* unfolding [46] are used to obtain Fat, Thin and Square matrices, respectively. The parameters for the Monte Carlo experiment are shown in Table 7.3.

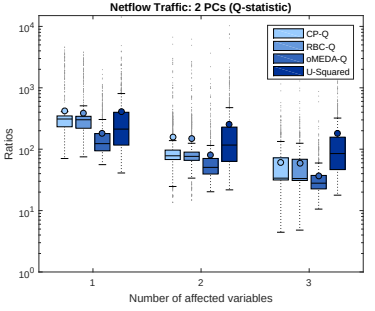
τ	N	M	$\mathscr{Y}$	pcs	$\mathscr{V}$
Thin (T)	3000	11	1	$\{1,2\}$	$\{1,2,3\}$
Square (S)	900	781	1	$\{1,2\}$	$\{1,2,3\}$
Fat (F)	30	1100	1	$\{1,2\}$	$\{1,2,3\}$

Table 7.3 Parameters involved in verification using *Saccharomyces cerevisiae* process data.

These data are altered in the same way as in the simulation section but using only one replicate for each considered type of calibration matrix (Thin, Square and Fat). The number of selected PCs is *i)* $pcs = 1$, and *ii)* $pcs = 2$, *i.*



(a) 1 PC



(b) 2 PCs

Fig. 7.9 Ratios in the Q-statistic, γ, for (a) 1 PC, and (b) for 2 PCs corresponding to the *Communications Network Traffic* data set.

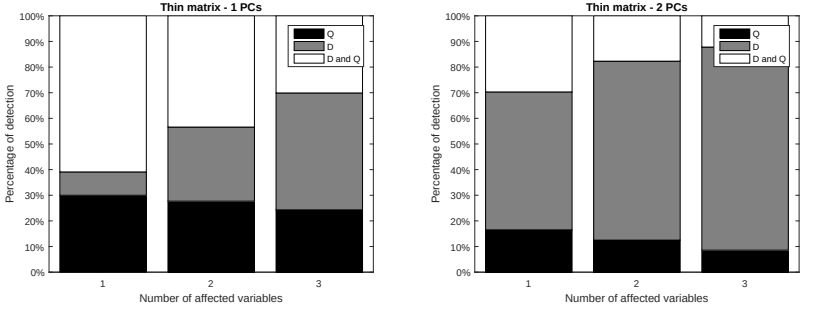
e., the number of PCs that captures the 75% of the total variance. The number of observations selected meets that in the calibration matrix. The variables are altered until any of the statistics is $K = 2$ times its control limit.

Results

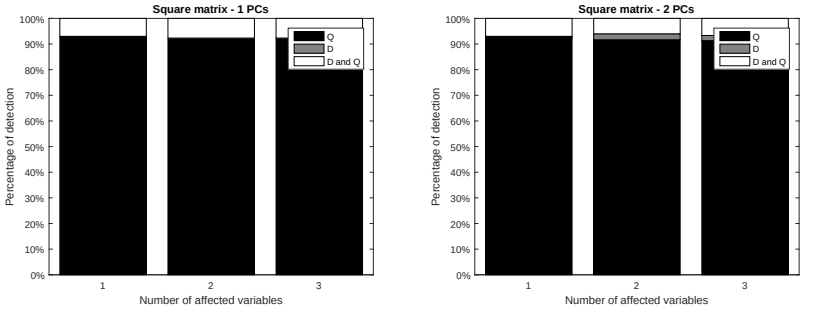
Fig. 7.10 shows the percentage of anomalies generated for each statistic. Compared to the distribution of probabilities obtained using synthetic data, the probability of generation only in the Q-statistic decreases for each type of matrix. There is a greater probability of generating an anomaly in both statistics simultaneously, compared to the results obtained for the data simulated with simuleMV.

In the same way as for the synthetic data, an ANOVA is performed on the ratio values. The study considers the same factors as in the first experiment: selected PCs (*pcs*), number of affected variables ($\mathscr{V}$), diagnosis method, type of matrix (τ), and statistics, as well as the first-order interactions. The ANOVA result is consistent with that obtained using simuleMV as it shows that all these factors and the corresponding interactions are statistically significant ($p - value < 0.01$).

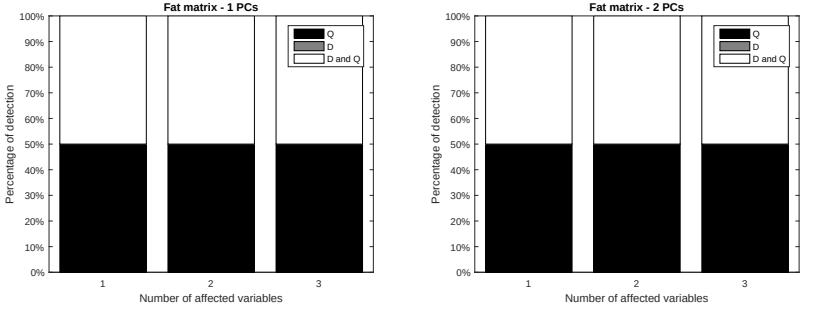
The most relevant parameters according to the effect size are the same as those in the simuleMV results: the type of matrix (τ), the statistic, and the diagnosis method. These parameters also present strong interactions. According to these results, the comparison is performed individually for each combination of statistic and type of matrix. The LSD plots are computed when there exist statistically significant differences among the approaches (see Fig.7.11). The results are coherent with those obtained for the synthetic data: *U-Squared* is in most cases better than the other methods, except for the Q-statistic for Square matrices, where CP and RBC are better. For the Q-statistic for Fat matrices, there is no significant difference among the methods.



(a) Thin matix (1000x11)

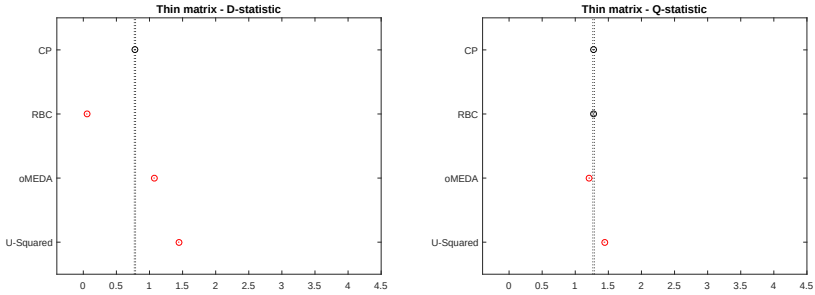


(b) Square matrix (300x781)

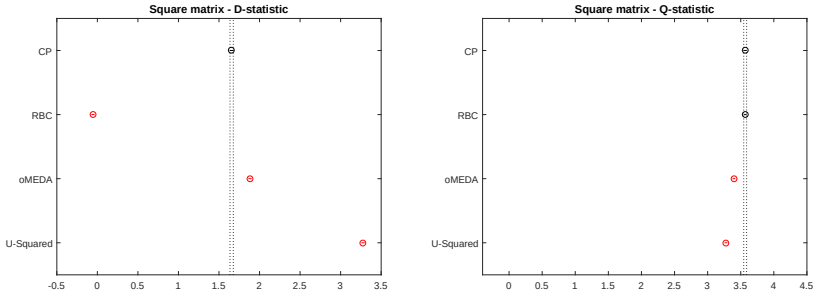


(c) Fat matrix (10x1100)

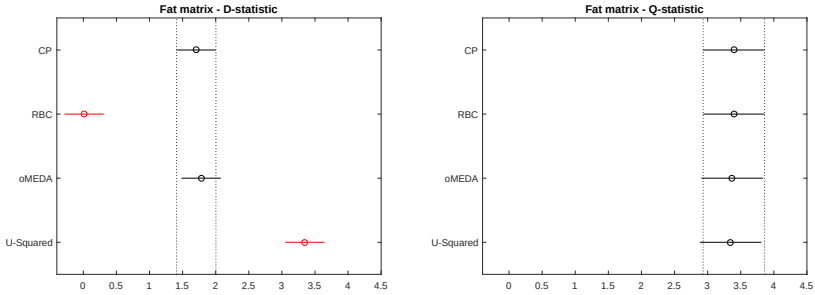
Fig. 7.10 Percentage of anomalies generated on each statistic for 1 PC and 2 PCs for (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) corresponding to the *Saccharomyces cerevisiae* process simulation.



(a) Thin matrix (300x11)



(b) Square matrix (900x781)

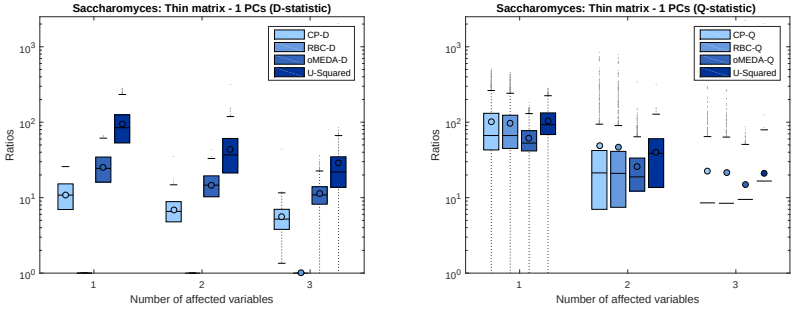


(c) Fat matrix (30x1100)

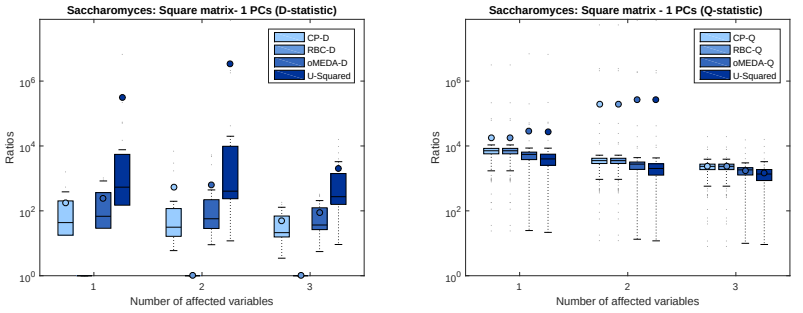
Fig. 7.11 LSD plots for (a) Thin matrices (T), (b) Square matrices (S), and (c) Fat matrices (F) corresponding to the *Saccharomyces cerevisiae* process simulation. The results for the D-statistic are shown in the left column, while the right column displays those results for the Q-statistic.

Fig. 7.12 and Fig. 7.13 show the distribution of the ratios computed after applying the diagnosis methods. The outcomes for anomalies detected in the D-statistic are on the left, whereas those for the Q-statistic are on the right. Note that if an anomaly is detected in both Q and D, it is included in both graphics. From a practical viewpoint, the differences are more relevant for the D-statistic, and similarly to the simulated data results, these differences are more evident when the number of PCs is increased. For the Q-statistic, the difference between *U-Squared* and the other methods is not significant.

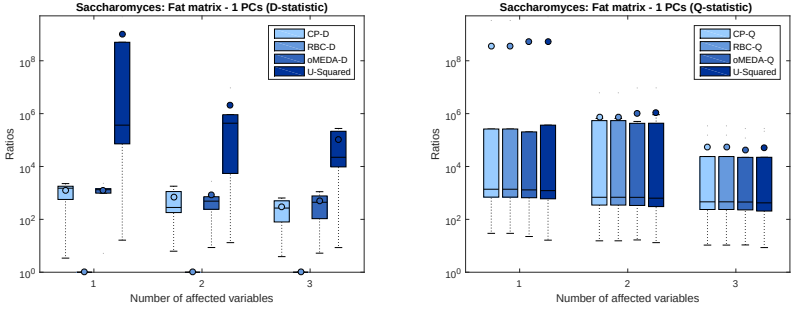
For this dataset, RBC does not show good results for the D-statistic either, and is useless for diagnosis when only 1 PC is selected.



(a) Thin matrix (1000x11) - 1 PC

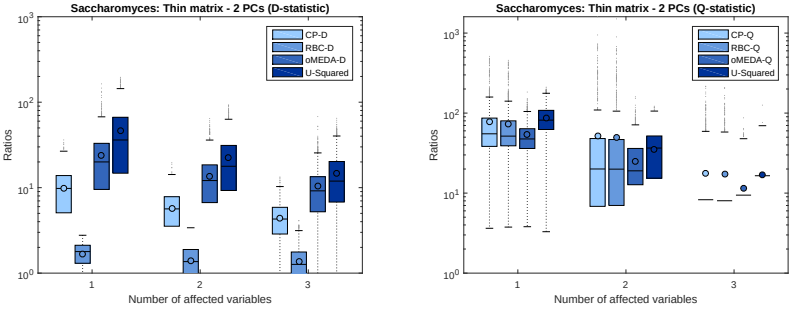


(b) Square matrix (300x781) - 1 PC

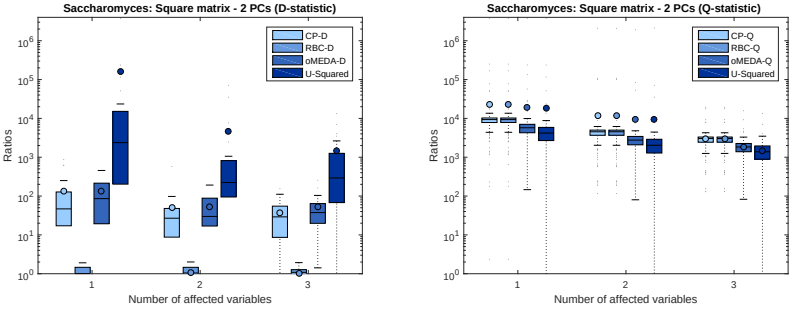


(c) Fat matrix (10x1100) - 1 PC

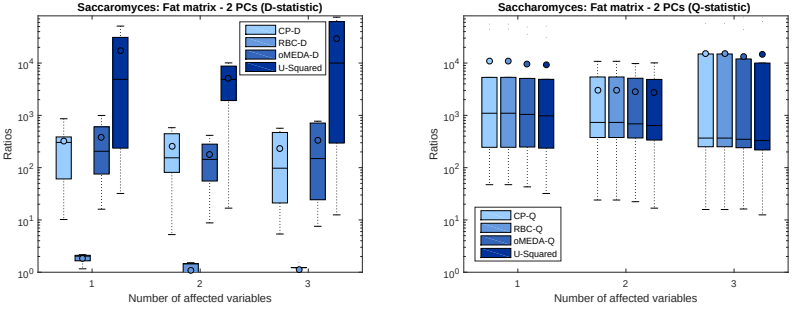
Fig. 7.12 Ratios, γ, for 1 PC for (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) corresponding to the *Saccharomyces cerevisiae* process simulation.



(a) Thin matrix (1000x11) - 2 PCs



(b) Square matrix (300x781) - 2 PCs



(c) Fat matrix (10x1100) - 2 PCs

Fig. 7.13 Ratios, γ, for 2 PCs for (a) Thin matrices (T), (b) Square matrices (S) and (c) Fat matrices (F) corresponding to the *Saccharomyces cerevisiae* process simulation.

7.5 Conclusions

In this chapter, a methodology to compare different diagnosis methods experimentally is presented. This methodology satisfies the requirements previously identified for a comprehensive comparison of diagnosis methods: *i)* anomalies with known diagnosis are generated, *ii)* a way to measure the diagnosis performance of each method is defined, *iii)* the parameters that might affect the diagnosis are identified, and an algorithm which integrates the previous requirements is proposed. This algorithm is integrated in a Monte Carlo procedure to obtain low uncertainty results.

This is a generic methodology, since the Monte Carlo approach allows the generation of anomalies that cover a wide range of possibilities, both univariate and multivariate and maintaining/breaking the correlation structure in the model. However, as already stated, the results should be interpreted considering that there is no theoretical warranty that all types of failure are covered.

Three state-of-the-art diagnosis methods of MSPC are compared using the proposed methodology: Contribution Plots (CP), Reconstruction-Based Contributions (RBC) and observation-based Missing-data method for Exploratory Data Analysis (oMEDA). A fourth method that follows a univariate approach is also included, Univariate Squared (U-Squared), with the following reasoning: when an anomaly occurs, the correlation structure in the model may not hold for such anomaly and therefore the division in model/residuals found for calibration data may not be optimum for diagnosis. In such case, one can consider that, for diagnosis, it makes sense to calculate the contribution of the variables to the full variables space instead of separated for each statistic. Applying oMEDA to the full variable space leads to derive the U-Squared expression and to include it in the comparison. The univariate diagnosis shows good performance results even when the correlation is not broken. The comparison is validated using realistic datasets from a communications network

and from a process simulation of production of *Saccharomyces cerevisiae*. These results are consistent with those obtained from the synthetic data.

This study leads us to propose a mixed PCA-MSPC/MSNM process in which detection is performed using a multivariate approach but diagnosis is performed via a univariate method.

8

MSNM Extensions Applied to Real Data

"No amount of experimentation can definitely prove that I am right; but a single experiment can prove that I am wrong."
Albert Einstein, Nobel Prize in Physics in 1921

"One never will see what she has done, but what there is still to do."
Marie Curie, Nobel Prize in Physics in 1903

This chapter evaluates the MSNM extensions presented in the following research papers using real network data:

- **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019b). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. *Journal of Chemometrics*, 33(11)

- **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2018). Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control. *Chemometrics and Intelligent Laboratory Systems*, 172:194–210

- Camacho, J., García-Giménez, J. M., **Fuentes-García, N. M.**, and Maciá-Fernández, G. (2019b). Multivariate Big Data Analysis for Intrusion Detection: 5 steps from the haystack to the needle. *Computers and Security (COSE)*, 87

- Maciá-Fernández, G., Camacho, J., García-Teodoro, P., and Rodríguez-Gómez, R. A. (2016). Hierarchical PCA-Based Multivariate Statistical Network Monitoring for Anomaly Detection. *International Workshop on Information Forensics and Security*

Contents

8.1 MSNM extensions 182
8.2 Materials and Methods 184
8.2.1 Anomaly Detection Assessment 184
8.2.2 UGR'16 dataset 185
8.2.3 MSNM application 191
8.3 Results for Standard MSNM 199
8.4 Results for Hierarchical MSNM 206
8.5 Comparison of Hierarchical and Standard Approaches 211
8.6 Conclusions 212

In the previous chapters, the MSNM is explained: from its origins in the MSPC theory to the contributions to the multivariate statistical monitoring in this thesis. The alternatives and extensions for the MSNM application are also described.

The last contribution of this PhD is the application of some of the MSNM extensions to a real case. The UGR'16 [130] dataset is selected for such purpose. This dataset contains a large capture of real network data that were collected from an ISP during 2016 (see Sections 5.2.1 and 8.2.2). UGR'16 also includes attacks that allow the evaluation of the capability of detection and diagnosis of the MSNM approaches.

Once the sensors have collected the raw data and the features are extracted, the data can be fused in different ways (see Section 4.1.2). For this reason, the evaluation has been split into two parts: on the one hand, the experiments performed for the standard fusion of the data, which considers different types of data fusion and pre-processing methods; and, on the other hand, the application of MSNM for the hierarchical fusion of the data (**H**). Although

the hierarchical fusion is not a contribution of this thesis *per se*, it is applied and evaluated for the first time to real network data in this work.

The rest of the chapter is organized as follows: first, the experiments are described as a part of the materials and methods section. Then, the 5-step methodology [36] (recall Chapter 4) is applied with two variants of the standard fusion of the data, allowing the evaluation of the extensions for the pre-processing and diagnosis. Then, the 5-step methodology is also applied for the hierarchical MSNM. To conclude, the results and conclusions of the evaluation are presented.

8.1 MSNM extensions

There exist extensions for each of the MSNM steps but for the parsing, which follows the original feature-as-a-counter approach [42]. The extensions applied and studied in this chapter are:

- **Fusion** step. Data can be joined to a single data matrix in different ways. We suggest two alternatives to obtain the *standard* fusion: *i) Concatenating* the value of the features (**C**); and *ii) Aggregating* the value of the features (**A**)[1]. Both are represented in Fig. 8.1.

- **Detection** step. This step is generally composed of a set of sub-steps, namely: the *pre-processing*, the *PCA model building*, and the *computation of statistics*. For network data, the pre-processing is usually done by Auto-Scaling (AS). This approach does not take into account the cyclo-stationarity of the data. Thus, we propose here to consider the cyclo-stationarity of network data, and for this reason we apply X-PARAMO (our proposal for BMSPC [192]) and TCS (one of the

[1] Please, remember from Chapter 4 that **A**-fusion can only be applied if the data sources are of the same type (*e.g.* all of them routers measuring `netflow`).

most extended methods for BMSPC). This work follows an unsupervised approach, thus we do not consider the optimized scaling [40], due to its semi-supervised nature makes it not comparable to the rest of methods in a fair way in the context of this work (unsupervised anomaly detection).

- **Diagnosis** step. Once an anomaly is detected in the previous step, it is needed to investigate what happened before the anomaly took place. To do this, there are many multivariate diagnosis methods that were explained in Chapter 7. Here we assess our proposal, U-Squared [195] as well as the method in which is based, oMEDA.

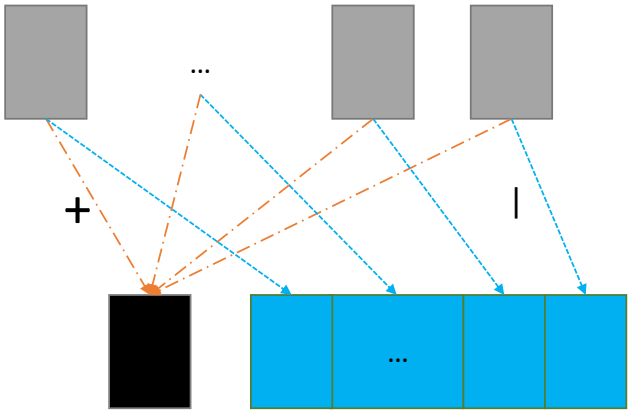


Fig. 8.1 Standard fusion: Aggregating (**A**-fusion) is denoted with '**+**'. Concatenating (**C**-fusion) is indicated with 'I'.

These extensions are applied following the 5-steps methodology [36], which is also an MSNM extension. In addition, we evaluate a second alternative to the *standard* fusion, the *hierarchical* organization of the data. The hierarchical approach was proposed as an MSNM extension for the first time in [129]. As explained in Chapter 4, there are different levels of data fusion: *low*, *middle* and *high* [61]. To evaluate this approach we have selected the

alternatives that (to the best of our knowledge) may lead to significant different results, considering those previously obtained for the evaluation of standard fusion (see Section 8.2.3). Thus, the hierarchical approach is assessed applying **C**-fusion, AS for pre-processing and U-Squared for diagnosis. Finally, sometimes, the hierarchical approach requires applying some re-ordering in the diagnosis and fusion steps in comparison to the original MSNM and the 5-steps methodology, which will be detailed in Section 8.2.3.

8.2 Materials and Methods

This section describes the metrics and the dataset used for the experiments, as well as the way of performing such experiments, including the fusion of the data and the variants applied of the MSNM steps.

8.2.1 Anomaly Detection Assessment

The relation between the number of false and true positives is very important in network monitoring, since the number of alarms may be excessive in this context. A way to measure this relation is the Area Under the Curve (AUC), which represents the area under the Receiver Operating Characteristics (ROC) curve [97, 138]. This is a typical measure for one-class classifiers, where the anomaly-based IDSs can be enclosed, and also for 2-class classifiers. An ideal classifier (a classifier that detects all the true positives without any false positives) has $AUC = 1$, while a classifier with poor capabilities for correct anomaly detection has $AUC \approx 0.5$, which is similar to a random classifier [8]. The type of classifier proposed in previous chapters for MSNM follows an unsupervised approach that, in some sense, can be considered to be a one-class classifier. For this reason, we use both AUC and ROC curves to assess the performance of the techniques under study in the context of the UGR'16 dataset.

8.2.2 UGR'16 dataset

Recall from Section 5.2.1 that the UGR'16 dataset is a network traffic data collection that was captured during a total of four months. During the fourth month, the traffic network was collected while a series of controlled attacks were launched in the same network. To insert the controlled attacks, a total of 25 virtual machines were installed in some of the sub-networks, with a similar configuration to that used in other ISP clients: 5 attackers referred to as A_1 to A_5, and 20 victims, referred to as V_{11}-V_{45}. Machines A_1 to A_5 attack the rest of the virtual machines (V_*) in different timestamps during a given period of time. Different types of attacks were implemented, which are labeled as **DoS**, **scan11**, **scan44** and **botnet**:

- **DoS** attacks are executed by sending `TCP SYN` packets to the victims using `hping3` with destination port 80. `hping3` is a command line Linux tool used to verify the connectivity between two machines. It works by sending some packets that can be modified [173, 174]. **DoS** traffic is merged with the real background traffic. Each packet has a size of 1280 bits with a rate of 100 packets/s.

- The **scan11** attack is a port scan, where a single attacker scans a single victim. In this type of attack it is common to find a high number of ports that are being scanned. These ports can be related to well-known services, such as 'messenger' or 'emule', but also to more unusual services.

- The **scan44** attack is another type of port scan, where four of the attackers simultaneously start a scan against victim machines. This type of attack works in the same way as **scan11** does (in relation to the features of the attack by itself). The difference is in the organization of the attackers and the victims.

- The **botnet** attack is a simulation of a `NERIS botnet`. A **botnet** attack consists on the infection of a number of machines by means of any type of malware to transform them in *bots* that are under the control of a *master* machine. After the infection, the **master** sends orders to the **bots**, which is usually a mechanism applied to be able to perform attacks in higher dimensions. Thus, the master needs to be connected with the bots to send them the instructions. This is frequently done using `IRC`, `p2p` or `HTTP`, among others [209]. The bot transmission can be synchronous (all the attacks are initiated by the attackers at the same time) or asynchronous (the attacks are initiated at different instants of time). Some of the most common types of attacks performed by a **botnet** are *Distributed Denial of Service (DDoS)*, *SPAM campaigns* or *Bitcoin mining*, among others. In the case of UGR'16, the traffic generated by a **NERIS** bot [79, 80] is adapted and inserted in the trace as if it were generated by the 20 victim machines. This permits to evaluate the behavior of a botnet infection scenario.

Fig. 8.2 shows two different days, a first day when the `DoS` attack took place (displayed in red), in contrast with the same day of the week obtained from the NOC data (represented in blue). This is an illustrative example of the implications of the attacks in relation to the NOC network traffic.

Finally, it is worth mentioning that the original capture also contains real anomalies that were properly labeled. Some of these anomalies correspond to `SSH` scan, SPAM, or `UDP` scan attacks. During the capture, an important anomaly was also found in the UGR'16 dataset during June (note that this anomaly was not even detected in the original paper that describes the dataset): a sudden increase in the `IRC` traffic between one of the virtual machines in the ISP network and an external IP (see Section 8.2.2). The existence of such anomalies is considered and dealt with when the experiments are performed to ensure the results are reliable.

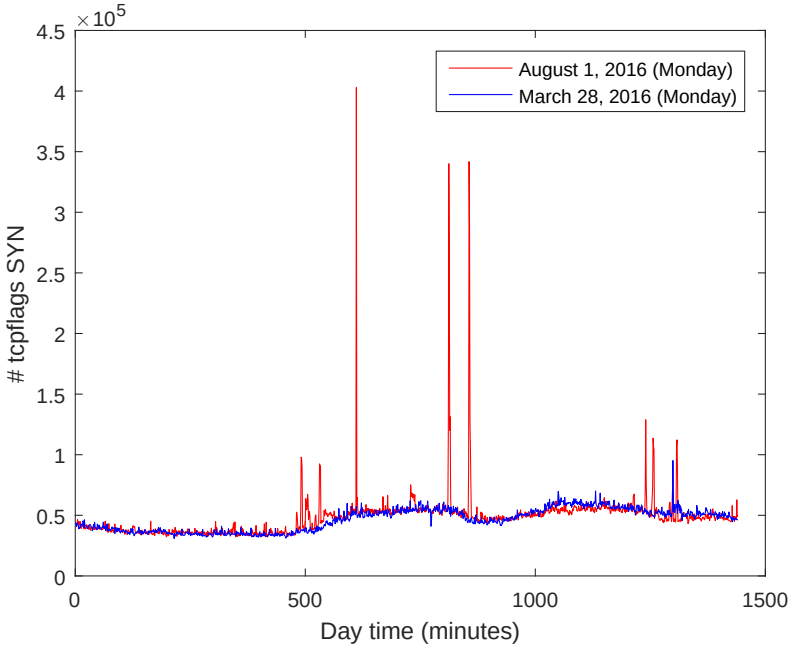


Fig. 8.2 Comparison in terms of `TCP SYN` counts between a day with a `DoS` attack (red) and an attack-free day (blue).

After applying Step 1 (parsing) and Step 2 (fusion) of MSNM, a NOC matrix, $\mathbf{X}$, (months from March to June) is obtained with $N = 136.180$ observations and a test matrix, **test**, (last month of the capture) with $N = 47.275$ observations. The number of variables is $M = 143$, where the last 9 are control variables, which are used as a Ground Truth. Thus, the number of monitoring variables is $M = 134$.

Separation in "Virtual" routers

Recall from Section 4.1 that MSNM follows a feature-as-a-counter approach [42]. To evaluate **C**-fusion (concatenation of the data from different sensors) and the **H**-fusion (hierarchical concatenation of the statistics and/or features ob-

tained in different sensors and layers) we need to split the traffic in different sensors. For this reason, the collected data are split into three different Virtual Routers (VRs) with sensors named *VR1*, *VR2* and *VR3*, and the connections are assigned to each virtual router depending on the IP addresses of the connection end points. When a connection has end-points belonging to several virtual routers, this connection is assigned only to one of these sensors. Table 8.1 shows the criteria followed to assign each connection to a virtual router. For each connection, the IP addresses (both source or destination) are checked: as soon as any of the IP matches an entry in the table, the connection is assigned to the corresponding virtual router (and sensor). As a result, we obtain three NOC matrices: $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ with the same dimensions as $\mathbf{X}$; and three matrices for the test data: $\mathbf{test}_1$, $\mathbf{test}_2$, and $\mathbf{test}_3$, also with the same dimensions as **test**. Recall that this dis-aggregation is possible due to the feature-as-a-counter approach [42], which also allows the future fusion in different forms.

Source or Destination IP	Virtual Router
42.219.156.0/24	*VR1*
42.219.152.0/24 or 42.219.154.0/24 or 42.219.158.0/24	*VR2*
Rest (including external)	*VR3*

Table 8.1 Criteria to assign end-points to virtual routers (with sensors).

Following these criteria, the attacks are distributed as shown in Table 8.2. Note that none of the attacks is captured by in *VR3* (see [130] for more details about the sub-network that each machine belongs to).

Real IRC Anomaly in UGR'16

Separating the data in virtual routers made it possible to find out an important anomaly during June: a sudden increase in the `IRC` traffic between one of the

Attack	Machines	VR1	VR2
DoS	$A_1 \rightarrow V_{21}$	-	✓
	$A_1 - A_5 \rightarrow V_{21}, V_{31}, V_{41}$	-	✓
scan11	$A_1 \rightarrow V_{44}$	-	✓
scan44	$A_1 \rightarrow V_{21}$	-	✓
	$A_3 \rightarrow V_{31}$	-	✓
	$A_4 \rightarrow V_{41}$	-	✓
	$A_2 \rightarrow V_{11}$	✓	-
botnet	V_*, A_*	✓	✓

Table 8.2 Distribution of the attacks in the virtual routers (with sensors).

virtual machines in the ISP network and an external IP address. This can be seen in Fig. 8.3.

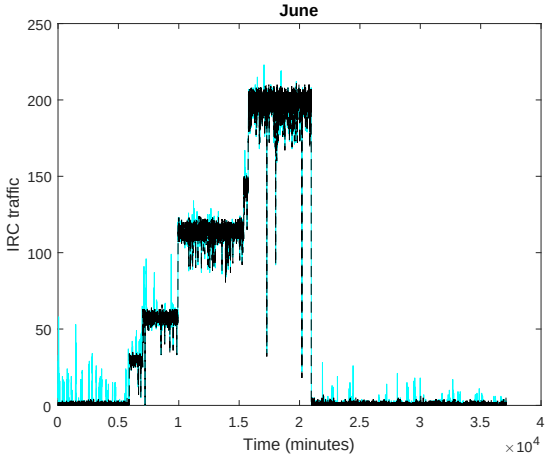


Fig. 8.3 IRC traffic in UGR'16 during June.

Before this finding, June was part of the calibration data and, thus, the anomalous traffic was considered to be normal. This turned in the deterioration of the general results in the detection, but specially of those related to the botnet attack. The reason was that having high IRC traffic is a key feature in many botnets, and this is exactly the case of the botnet that was introduced

for testing in the UGR'16 dataset, and that we need to detect [40]. The separation performed in the previous section was decisive to discover that the `botnet` was properly detected in *VR2*, but not in *VR1*. This was possible thanks to the diagnosis (see Fig. 8.4) carried out as a part of the Phase I of the application of the MSNM methodology. The attack in the test data was not detected in *VR1* because there was an abnormally high `IRC` traffic in this router during the calibration.

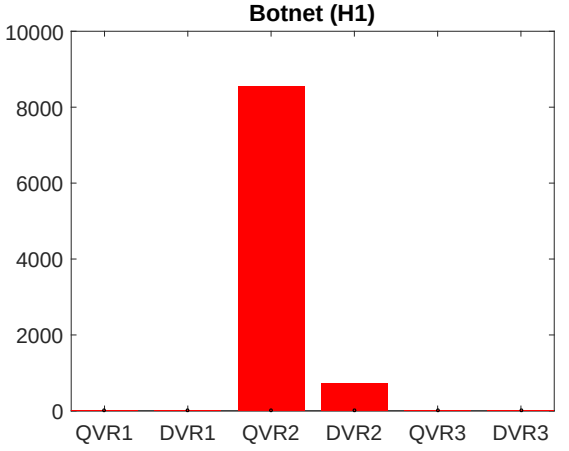


Fig. 8.4 Wrong diagnosis of the `botnet` (model calibration includes anomalous `IRC` traffic in June).

After that, a *de-parsing* process was carried out, looking in the original records (take as an example Fig. 8.5). The analysis unveiled that, indeed, there were about 3.8M of connections between `42.219.156.231` (virtual machine in *VR1*) and `168.227.46.37` (external machine, *VR3*) during several days of June. This represents the 98.5% of the connections, which transmitted 1.3GB of data. The external machine (`168.227.46.37`) varied the source and destination ports, while the local machine (`42.219.156.231`) always used the port `6667`. Once the anomaly was confirmed, June was removed from the calibration data in all the studies.

```
Top 10 IP Addr ordered by flows:
Date first seen          Duration Proto          IP Addr    Flows(%)      Packets(%)       Bytes(%)
2016-06-01 11:15:19.296 1931760.236 any      42.219.156.231    3.8 M(98.5)   18.7 M(96.7)    1.3 G(89.5)
2016-06-05 02:10:29.032 904403.568 any       168.227.46.37    3.8 M(98.5)   18.7 M(96.6)    1.3 G(89.4)
2016-06-01 00:05:13.556 490637.232 any       188.31.202.87     9098( 0.2)     9098( 0.0)   400048( 0.0)
2016-06-01 05:58:50.244 457199.364 any        60.99.59.45      8978( 0.2)     8979( 0.0)   394800( 0.0)
2016-06-05 01:36:27.616 906488.452 any       199.95.74.246     3403( 0.1)    10418( 0.1)    1.3 M( 0.1)
2016-06-01 00:14:45.080 2225408.748 any       42.219.159.90     3280( 0.1)     6930( 0.0)   390306( 0.0)
2016-06-01 00:49:02.072 2214303.500 any       42.219.159.85     2229( 0.1)    29414( 0.2)    7.0 M( 0.5)
2016-06-01 07:15:14.568 1117254.500 any      170.228.7.189      2049( 0.1)     2054( 0.0)    90208( 0.0)
2016-06-01 00:51:56.240 2218830.696 any       42.219.155.28     1745( 0.0)    17861( 0.1)    4.7 M( 0.3)
2016-06-02 22:17:28.752 1298513.612 any      218.12.37.188      1332( 0.0)     1399( 0.0)    61584( 0.0)

Top 10 Port ordered by flows:
Date first seen          Duration Proto              Port    Flows(%)      Packets(%)       Bytes(%)
2016-06-01 00:05:13.556 2226066.888 any             6667    3.9 M(100.0)   19.4 M(100.0)    1.5 G(100.
2016-06-05 02:10:29.032 904403.568 any             59299    19741( 0.5)    35181( 0.2)    2.7 M( 0.2)
2016-06-01 00:18:37.576 2225262.868 any            49717     8875( 0.2)     9366( 0.0)   393457( 0.0)
2016-06-01 00:12:07.488 2225566.340 any              445     4389( 0.1)     9482( 0.0)   454830( 0.0)
2016-06-01 00:21:50.248 2223812.828 any               80     3833( 0.1)   101959( 0.5)    88.8 M( 5.9)
2016-06-01 00:19:11.188 2224551.860 any               53     3553( 0.1)    10838( 0.1)   866195( 0.1)
2016-06-05 01:36:27.616 906488.452 any             55146     3403( 0.1)    10418( 0.1)    1.3 M( 0.1)
2016-06-01 00:42:24.716 2219402.220 any              443     2728( 0.1)    30649( 0.2)    12.1 M( 0.8)
2016-06-15 10:26:39.568 28214.464 any              27435     1175( 0.0)    11745( 0.1)    3.3 M( 0.2)
2016-06-20 13:13:29.212 12255.884 any              27319      451( 0.0)     4840( 0.0)    1.6 M( 0.1)

IP addresses anonymised
Summary: total flows: 3853631, total bytes: 1494795033, total packets: 19394543, avg bps: 5371, avg pps:
Time window: <unknown>
Total flows processed: 3470901813, Blocks skipped: 0, Bytes read: 180479735636
Sys: 187.488s flows/second: 18512574.6 Wall: 233.198s flows/second: 14803902.7
```

Fig. 8.5 Statistics by IP and port (output from the nfdump tool).

8.2.3 MSNM application

The calibration dataset includes traffic from $2016/03/19-00:00$ to $2016/05/31-23:59$[2]. That is, more than two months. Note that the capture starts at $2016/03/18-10:52$ and finishes at $2016/06/26-18:27$. The observations at the beginning of the dataset are removed to start with the first complete day, and all the observations corresponding to June are also removed due to the IRC anomaly that was found during initial explorations (see Section 8.2.2). Additionally, the anomalies labeled in [130] are also removed to have a NOC dataset for the calibration; yielding $N = 98.262$ observations, corresponding to minutes.

For the test data we only consider the anomalies that correspond to synthetic attacks. In the same way as for the calibration, the days that are not complete are removed from the dataset. In addition, only those containing the artificial attacks are considered. Despite the fact that in the UGR'16

[2]Note that this is expressed in American format.

paper the authors explain that synthetic attack batches are spanned during 12 days, during the experiments it was discovered that the last two phases of the `botnet` (corresponding to the two last days of attacks) are not in the data: one of them owing to a *parsing* error, and the other due to a miss insertion in the raw data. For this reason, the test dataset contains only 10 days, going from $2016/07/28-00:00$ to $2016/08/06-23:59$, yielding $N = 43.200$ observations.

Since the experiments are intended to evaluate our proposals for the pre-processing (Chapter 6) and the diagnosis (Chapter 7), Steps 1 and 2 of the MSNM methodology were applied before starting this work and they are not shown here. Step 3 (detection) is performed as follows:

1. *Pre-processing*. The data are pre-processed by Auto-Scaling (AS) them.

2. *Model building*. The PCA model is created following Equations (3.3) to (3.5). For this purpose, the Big Data functionality in the MEDA-Toolbox [51] is used to build the model applying an iterative update. For the PCA model, 1 PC is selected, since it captures most part of the variance.

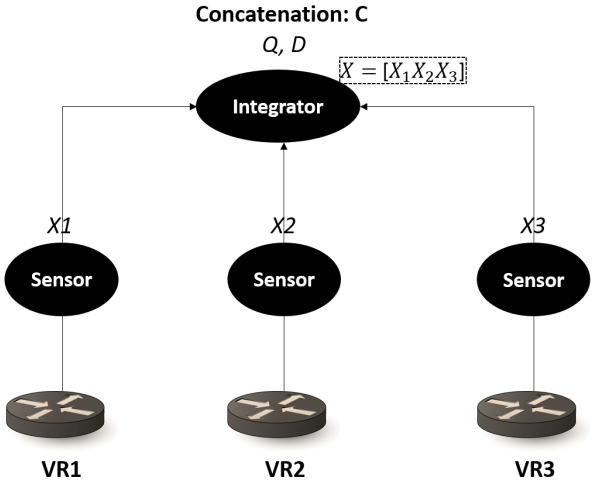
3. *Compute the statistics*. The statistics and their corresponding control limits in Phase I are calculated following Equations (3.6) to (3.9) and (3.12), which is also performed using the Big Data functionality in the MEDA-Toolbox. Then, the Equation (4.1) is applied in Phase II for detecting the anomalies (attacks).

The diagnosis (Step 4) is performed by applying U-Squared [195] and oMEDA [29]. Finally, as a part of the 5-steps methodology evaluation, the de-parsing (Step 5) is manually performed: once a set of observations is detected as anomalous and the diagnosis signals a number of related features, the raw records are inspected, by making queries with the constraints obtained in the detection and the diagnosis steps.

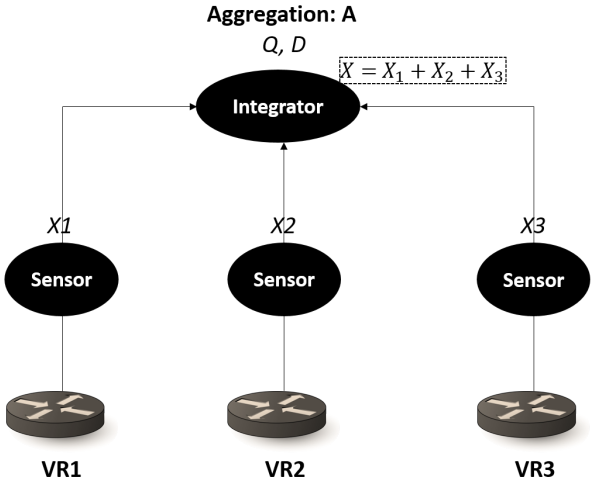
Standard MSNM

A standard MSNM means in this context that the data are combined into a single matrix using a single low-level fusion [61, 67, 180]. In this scenario, the fusion of the data from all the sensors can be performed in two different possible configurations: *i)* **A**-fusion, $X = X_1 + X_2 + X_3$ (see Fig. 8.6 (a)); and *ii)* **C**-fusion, $X = [X_1\ X_2\ X_3]$ (see Fig. 8.6 (b)). **A**-fusion consists in aggregating the data by columns (features). This is performed by summing the values of each feature and observation across the different data sources. Thus, in the case of the UGR'16 data set, where 134 features are defined (remember Section 8.2.2), the number of features would remain the same if we use the **A**-fusion, that is, $M = 134$. **C**-fusion consists in concatenating the data horizontally. This is performed by appending at the end of a matrix (after the last column) the matrix corresponding to the next data source, yielding a new matrix with the same number of observations. The number of features is the sum of the features of all the considered data sources. Therefore, for the **C**-fusion, the number of features would be $M = 3 \times 134 = 402$.

These experiments also involve the evaluation of different pre-processing approaches, including those that take into account the cyclo-stationarity of the data. These methods are usually applied in batch MSPC and were already validated in Chapter 6, using process data. Thus, we select *TCS* and *X-PARAMO* as pre-processing methods to consider the cyclo-stationarity of the data during the evaluation with real network data. X-PARAMO is applied with the three configurations shown in Table 8.3. These configurations are selected following the recommendations in Chapter 6: not to apply an excessive smoothing to avoid the information loss. In this case, the number of observations and the pace of the data are much more higher than for industrial processes. For this reason, we consider sizes of windows that span from seven minutes (a really small window in this context) up to one hour (a larger but yet reasonable



(a)



(b)

Fig. 8.6 Standard topology for (a) **C**-fusion and (b) **A**-fusion.

window). The forgetting factor help us to control the smoothing. In this case, $\lambda = 0.8$ only penalizes the observations most distant in time.

Setting	W	λ
s1	7	0.8
s2	21	0.8
s3	61	0.8

Table 8.3 Configuration settings for X-PARAMO. W represents the total size of the window (given in minutes), and λ corresponds to the forgetting factor.

Following the pre-processing methods from the BMSPC requires:

1. *Transforming the data* from two to three dimensions. This process yields a 1440*x*134*x*70 matrix, where 70 are the number of collected days, 1440 are the minutes in a single day, and 134 is the number of features (note that this does not changes in relation to the two-way matrix).

2. *Pre-processing* the three-dimensional data using the two selected methods and corresponding configurations.

3. *Unfolding the data* in a **variable-wise** mode. Note that this unfolding does not change the original arrangement of the data, since the N observations still correspond to the sampling time points. We discard **batch-wise** unfolding due to the extremely high number of observations of these data, which would lead to a 70*x*192.960 matrix.

Hierarchical MSNM

The hierarchical fusion of the data implies a number of benefits: the reduction in the volume of traffic towards the monitoring system, the scalability of the architecture, and a higher level of privacy of the monitoring approach [129]. At the same time, it maintains the main advantages of the **C**-fusion: the

location of the source of the IT alerts thanks to the diagnosis. An important concern when applying this approach is whether the performance is reduced in relation to the standard MSNM due to the effect of the different layers imposed by the hierarchy. To study the hierarchical MSNM, four different scenarios are investigated. The main differences among them are: the number of layers in the hierarchy; the position of the sensors to collect the data and build the models; and the way of creating the data matrix to build the models on each layer. For the model building, the MSPC Phase I is applied: the data are first pre-processed, then the PCA model is applied and, finally, the outliers are treated properly.

Scenario I (H1). This scenario has two layers: a *ground layer*, corresponding to the virtual routers and their individual sensors (*leaf sensors*), and a *top layer*, corresponding to the level where the statistics to be monitored are computed. Here, a different model is computed on each of the leaf sensors. Then, the statistics are also computed in these sensors, and they are sent to the top layer to be concatenated in a matrix, $X = [Q1\ D1\ Q2\ D2\ Q3\ D3]$. The result is an $Nx6$ matrix, which is used to build the model in the top layer (see Fig. 8.7 (a)). The fusion performed by the integrator at the top layer is a high-level fusion [61, 180], since it fuses the statistics, which are the output of the sensors.

A second layer is added to the hierarchy with the aim of evaluating this type of fusion with additional layers and for different organizations of the routers. We decided to include this additional layer to fuse two of the virtual routers. Then, in the top layer, the statistics resulting of this layer are combined with those obtained for the router not combined yet. Thus, we obtain three different scenarios by interchanging the different routers so that all of them are considered in each of the possible situations and we can study the effect of the different organizations in the performance of anomaly detection. These scenarios are described in the following paragraphs.

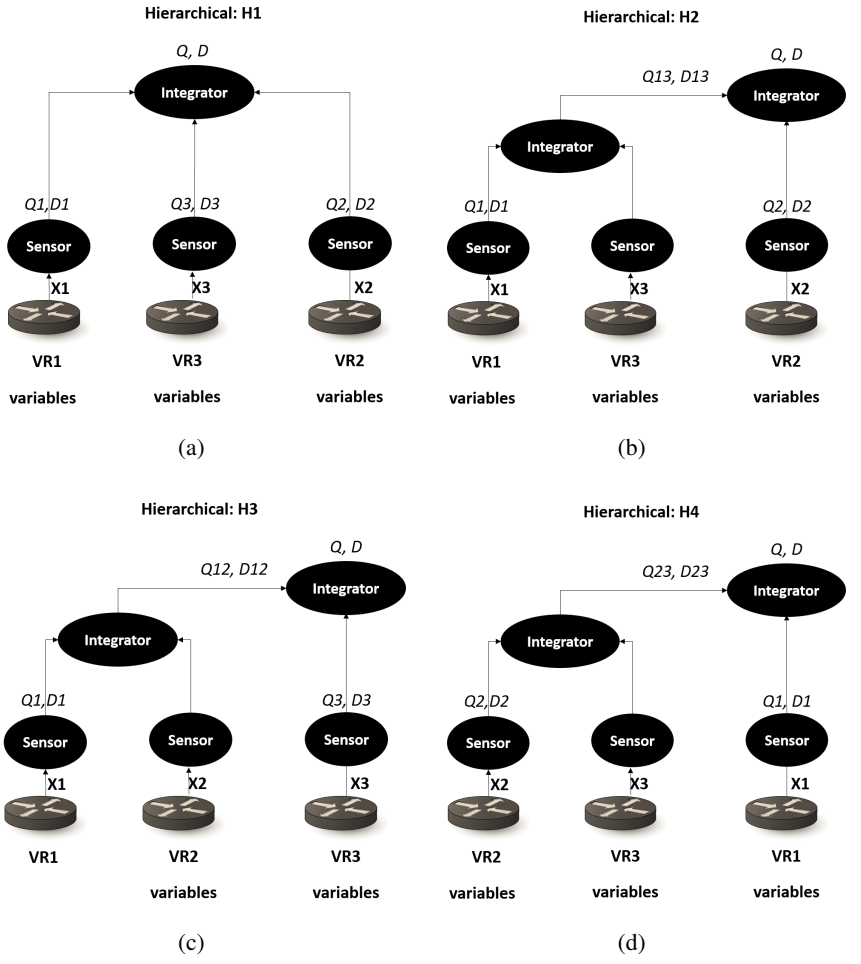


Fig. 8.7 Hierarchical topologies (a) **H1**, (b) **H2**, (c) **H3**, and (d) **H4**.

Scenario II (H2). This scenario has three layers: a *ground layer*, corresponding to the virtual routers and their individual sensors (*leaf sensors*), a *second layer*, where an intermediate sensor collects data from VR1 and VR3, and a *top layer*, corresponding to the layer where the statistics to be monitored are computed. An individual model is built for VR1, and the cor-

responding statistics are computed ($Q1$ and $D1$). These statistics, together with the observations for the features in VR3 are integrated in the second layer, yielding $X13 = [Q1\ D1\ X3]$, which is an $Nx(2+M)$ matrix. The fusion performed by the integrator in the second layer is a combination of low- and high-level fusion [61, 180], since it fuses the statistics (which are the output of the sensor in VR1, high-level) with the features in VR3, low-level. We call this combination **hybrid-fusion**. A new model is built from $X13$, and the statistics are also computed. An individual model is also built in the leaf sensor for VR2. Then, the corresponding statistics are computed ($Q2$ and $D2$). These statistics, together with the statistics computed for the second layer, $Q13$ and $D13$, are collected at the top layer, yielding $X = [Q13\ D13\ Q2\ D2]$, which is an $Nx4$ matrix. Finally, at the top layer, a model is built from X (see Fig. 8.7 (b)). The fusion performed by the integrator at the top layer is a high-level fusion [61, 180], since it fuses the statistics, which are the output of the sensors.

Scenario III (H3). This scenario is the same as **H2**, interchanging VR2 and VR3 (see Fig. 8.7 (c)).

Scenario IV (H4). This scenario is the same as **H2**, interchanging VR2 and VR1 (see Fig. 8.7 (d)).

Since the effect of TCS and PARAMO has already been evaluated in Section 6.7, and there are no attacks affecting the cyclo-stationarity in the UGR'16 dataset, in this part of the chapter the pre-processing is performed with AS. The same applies to the diagnosis methods, which are evaluated in Chapter 7. Thus, *U-Squared* is applied to simplify the diagnosis. Note that, for the hierarchical approach, the diagnosis allows us to identify and prioritize the data source where an anomaly was originated. This is an added value to the original functionality of the diagnosis, which enables the identification of

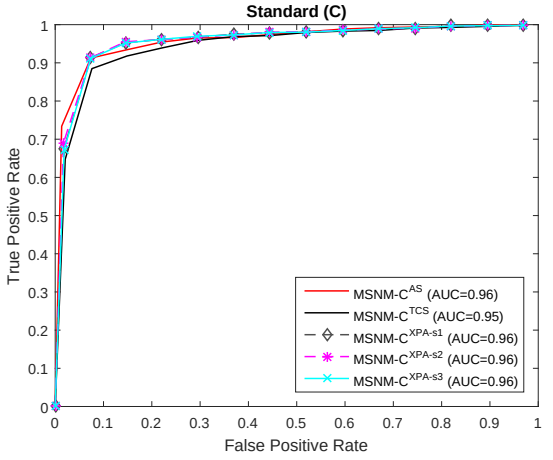
the topological location of the abnormal variables for a given anomaly, thus helping to discover the root cause of the anomaly.

8.3 Results for Standard MSNM

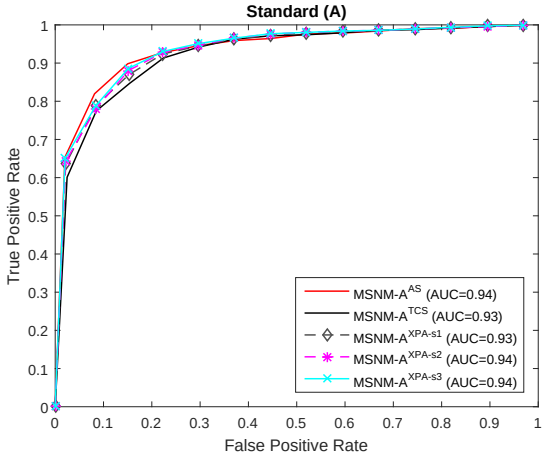
Detection (Step 3). As a first part in the study, an evaluation of the general detection capacity of the monitoring variants is performed, which is assessed using the ROC curves and the AUC. Fig. 8.8 represents the ROC curves for the different standard fusions: **A** and **C**, applying different pre-processing methods: AS, TCS, and X-PARAMO with the configurations shown in Table 8.3 (XPA-s1, XPA-s2, XPA-s3). All the evaluated alternatives present a high performance ($AUC \geq 0.93$). Furthermore, it can be observed that TCS and PARAMO present an equivalent AUC to the auto-scaling (the reference method). Finally, we can observe that the **C**-fusion shows a greater AUC than the **A**-fusion. This difference may be caused by the existing correlation between the features in the different sensors, which is captured by the model in **C**-fusion, but it is missed when using **A**-fusion.

As a second part of the study, the aforementioned alternatives are analyzed to assess the individual performance considering the type of attack. Fig. 8.9 shows the AUC grouped by attack type. It can be observed that the `scan44` is almost perfectly detected by all the evaluated alternatives. The `DoS` is the most difficult attack to detect, specially for the **A**-fusion methods. However, the performance is still $AUC \approx 0.9$ for this attack.

Diagnosis (Step 4). After the detection step, the anomalous observations that correspond to the attacks in the test dataset are diagnosed. As an example, Fig. 8.10 shows the U-Squared diagnosis for one sample of every attack type. We can observe that for the `DoS` and the *Botnet* attacks (Fig. 8.10 (a) and (d), respectively) some features are clearly signaled and the diagnosis is simple. However, for the two scan attacks (Fig. 8.10 (b) and (c)), identifying a clear



(a)

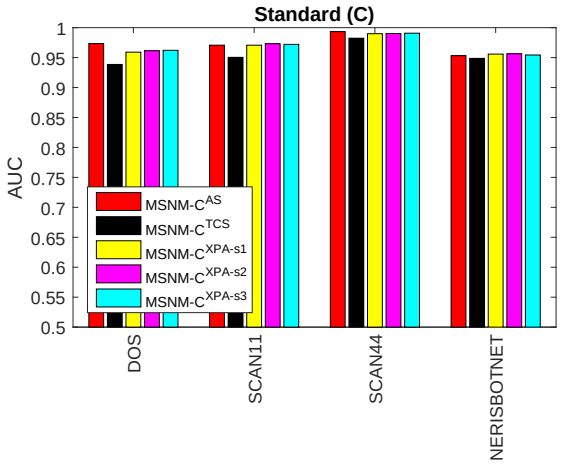


(b)

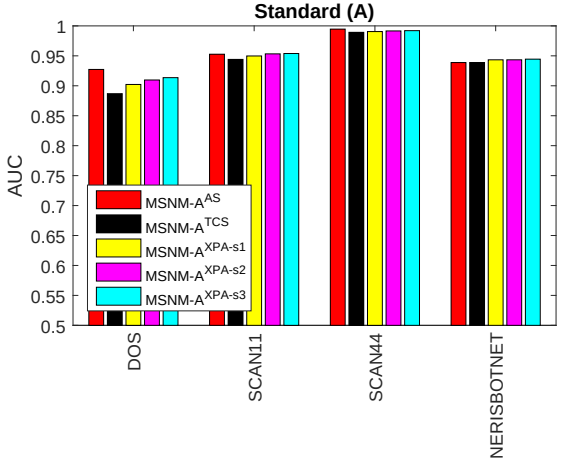
Fig. 8.8 ROC curves and AUCs for the standard fusion (a) **C** and (b) **A**.

number of variables is challenging[3]. In such cases, we recommended sorting

[3]Note that, for `Scan44`, the variables that are highlighted correspond to VR1. However, the attack affects both VR1 and VR2 (recall Table 8.2). This does not means that the anomalies



(a)



(b)

Fig. 8.9 AUC grouped by attack for the standard fusion (a) **C** and (b) **A**.

the variables according to their absolute value and in descending order as

are not diagnosed in VR2, this means that VR1 should be prioritized in relation to VR2, since the magnitude of the anomaly is higher in VR1.

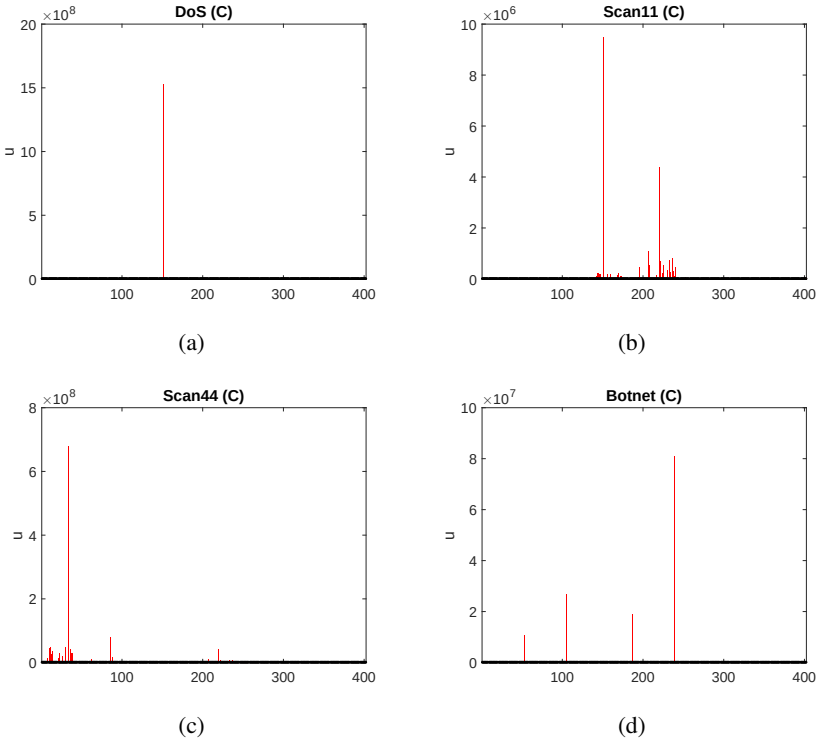
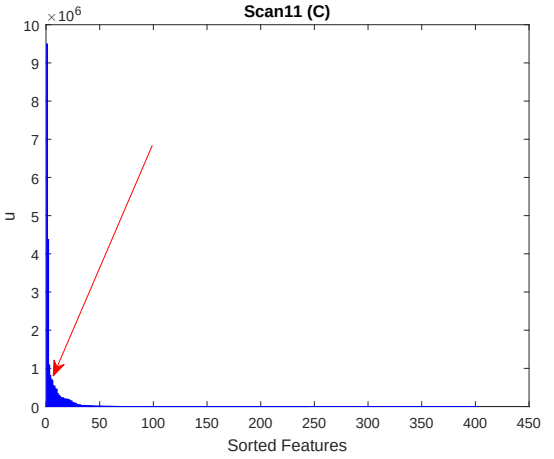


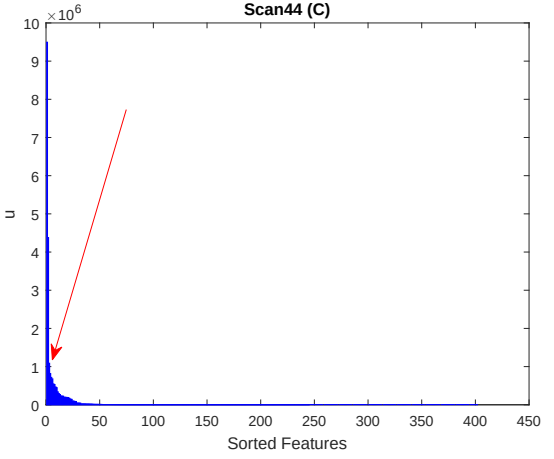
Fig. 8.10 U-Squared diagnosis applying **C**-fusion for (a) `DoS`, (b) `Scan11`, (c) `Scan44`, and (d) `Botnet`.

displayed in Fig. 8.11 and then to select those that are before the 'knee' of the plot.

We also compare in Tables 8.4 and 8.5 the most relevant features for the attacks after the diagnosis in **A**- and **C**-fusion for the U-Squared and oMEDA. Features in both tables are selected following the aforementioned 'knee' criterion. The diagnosis for all the attacks but the `DoS` in Table 8.5 are coherent with what we expected. We find surprising that `telnet` traffic is the most relevant variable for the Denial of Service (DoS) attack signaled for both diagnosis methods, since this feature is not specially related to this



(a)



(b)

Fig. 8.11 Sorted U-Squared diagnosis applying **C**-fusion for (a) `Scan11` and (c) `Scan44`. It is useful to select those features before the 'knee'.

DoS attack. For this reason, we tried to go deeper in the research of this attack, by investigating the raw data. Visualizing the telnet flows from **X** and **test** we observed that, indeed, there is a generalized increase in this type of traffic during the month of the capture for the test data. This anomaly is much more evident in VR2. Thus, the diagnosis is actually correct and the methods allow to detect an unexpected anomaly, which is clearly prioritized in the case of U-Squared, since the magnitude of this anomaly is higher than the magnitude of our controlled attacks. When oMEDA is applied for the botnet there are some features that are signaled but do not correspond to such attack (denoted with '**'): dport_dns, protocol_udp and sport_dns. We have investigated them and all show a lower value than in the calibration data, which explains that oMEDA signals them as anomalous.

Attack	U-Squared	oMEDA
DoS	sport_telnet*	sport_telnet*, sport_http, dport_http
Scan11	dport_kpasswd, sport_telnet*, dport_gopher, dport_citrix dport_msnmessenger	sport_telnet*, dport_citrix, dport_msnmessenger
Scan44	dport_kpasswd, dport_gopher, dport_citrix, dport_msnmessenger	dport_citrix, dport_msnmessenger, dport_register,dport_kpasswd
Botnet	dport_irc, sport_irc,	dport_irc, dport_dns**, protocol_udp**, sport_dns**, sport_irc

Table 8.4 Diagnosis for **A**-fusion applying U-Squared (univariate) and oMEDA (multivariate). Only the features with the highest value are displayed. Incorrect diagnosis are underlined. The '*' denotes that this is not considered an incorrect diagnosis, since we previously verified that, indeed there exists anomalous telnet traffic in the background. The same applies for '**'

Table 8.5 shows the results for the diagnosis after applying **C**-fusion. The *Scan44* attack is detected by oMEDA in VR2 while U-Squared signals VR1 for the **C**-fusion. Both methods provide a diagnosis that is partially correct, since the attack takes place both in VR1 and VR2. The main feature for the *Botnet* attack, the IRC traffic, is signaled properly by both diagnosis methods.

Besides, U-Squared identifies the two sources of the anomaly, VR1 and VR2. However, oMEDA only identifies the features from VR2 and also signals other features from VR3 (remember that there are no attacks coming from VR3). We think that this may be caused by the *smearing* effect. Note that U-Squared also signals `dport_telnetVR3` as anomalous (this is marked with an '*' in Table 8.5). However, this is not considered an incorrect diagnosis, since we previously verified that, indeed, there exists anomalous `telnet` traffic in the background.

Attack	U-Squared	oMEDA
DoS	`sport_telnetVR2*`	`tcpflags_RSTVR2`, `sport_telnetVR2*`, `sport_httpVR2`, `dport_httpVR2`
Scan11	`sport_telnetVR2*`, `dport_kpasswdVR2`, `dport_gopherVR2`	`dport_registerVR2`, `dport_citrixVR2`, `tcpflags_RSTVR2`,`sport_registerVR2`
Scan44	`sport_kpasswdVR1`, `dport_kpasswdVR1` `sport_quoteVR1`	`dport_citrixVR2`, `dport_registerVR2`, `dport_msnmessengerVR2`, `sport_registerVR2`
Botnet	`dport_ircVR2`, `dport_ircVR1`,	`sport_ircVR2`, `dport_ircVR2`

Table 8.5 Diagnosis for **C**-fusion applying U-Squared (univariate) and oMEDA (multivariate). Only the features with the highest value are displayed. The '*' denotes that this is not considered an incorrect diagnosis, since we previously verified that, indeed there exists anomalous `telnet` traffic in the background.

De-Parsing (Step 5). Finally, looking into the raw records, a deeper analysis was performed by using nfdump for querying the raw data. Let us take as an example the diagnosis of the `botnet` attack. One of the first things to be checked is whether there exist differences between a day with a `botnet` attack and a day free of attacks or not. To do this, we apply the following nfdump commands:

`nfdump -R`
`nfcapd.201607280000 : nfcapd.201607282359`
`-s ip 'src port 6667 or dst port 6667'`
for a normal day (the 28th of July).

`nfdump -R nfcapd.201608100000:nfcapd.201608102359 -s ip 'src port 6667 or dst port 6667'`
for a day affected by the `botnet`. In both cases, source and destination ports are filtered by port 6667, which in this case corresponds to the `IRC` port. The results confirmed that there was an abnormal increase in the `IRC` traffic at `2016/07/28-00:00`. Then, a manual de-parsing was carried out by performing similar queries, considering only those raw records corresponding to the anomalous observations and the diagnosed variables to build the nfdump queries. This was repeated for the rest of the days and attacks, varying the conditions of the queries to inspect the relation between the collected data, the diagnosed features, and the attacks.

8.4 Results for Hierarchical MSNM

Detection (Step 3). Fig. 8.12 represents the ROC for the different hierarchical scenarios: **H1** (2 layers), **H2** (3 layers), **H3** (3 layers) and **H4** (3 layers). All the evaluated alternatives, except **H4**, present a high performance ($AUC \geq 0.95$). The AUC for **H4** is much lower than for the rest of the scenarios.

Fig. 8.13 shows the AUC grouped by attack, where it can be observed that the `scan44` is almost perfectly detected by each of the evaluated alternatives. The `botnet` also present good results in all the cases. The `DoS` and the `scan11` have an $AUC \geq 0.95$ for **H1** to **H3**, while they are difficult to detect by **H4**, specially the `scan11` attack, which presents poor results for this scenario.

We explore two ways to improve **H4**:

- Creating the model for the second layer from the statistics, instead of using the variables from VR3, see Fig. 8.14 (a) and compare with Fig. 8.7 (d). This means that $X23 = [QVR2 \quad DVR2 \quad QVR3 \quad DVR3]$. The top layer is built in the same way as we did for **H4**: the statistics

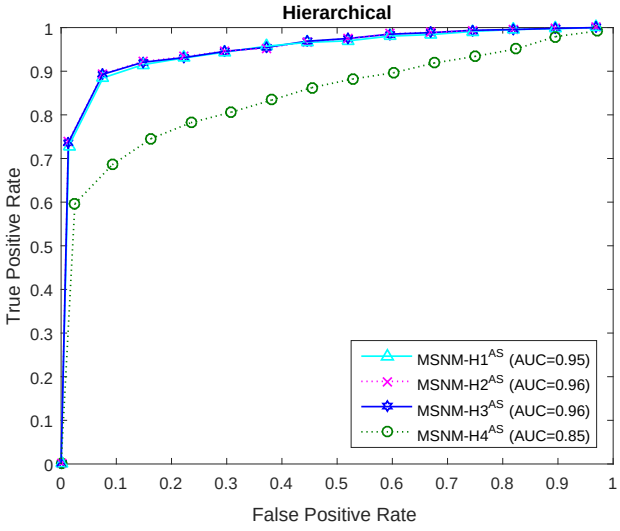


Fig. 8.12 ROC curves for the hierarchical fusion.

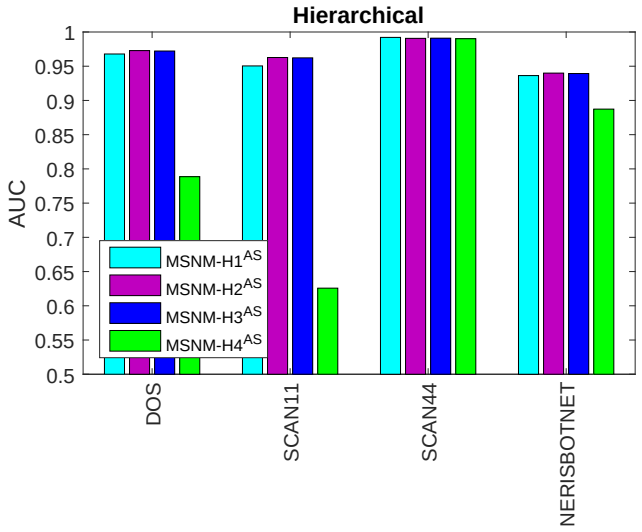


Fig. 8.13 AUC grouped by attack for the hierarchical fusion.

from the second layer are concatenated with the statistics calculated in VR1. Then, the PCA model is created. This scenario is called **H4b**.

- Weighting the features using block scaling (see Fig. 8.14 (b)): for the second layer, each of the statistics of VR2 are weighted after the pre-processing with a 0.5 factor, while the variables of VR3 are weighted with a $1/M$ factor. Thus, we have two blocks: one for VR2, and another for VR3, which are equally weighted. Thus, each branch of the hierarchy has the same importance for the model building in the top layer of the hierarchy. This scenario is called **H4c**.

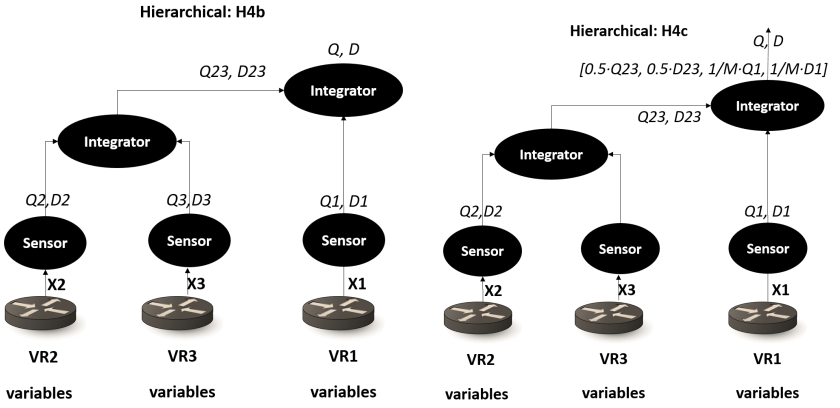
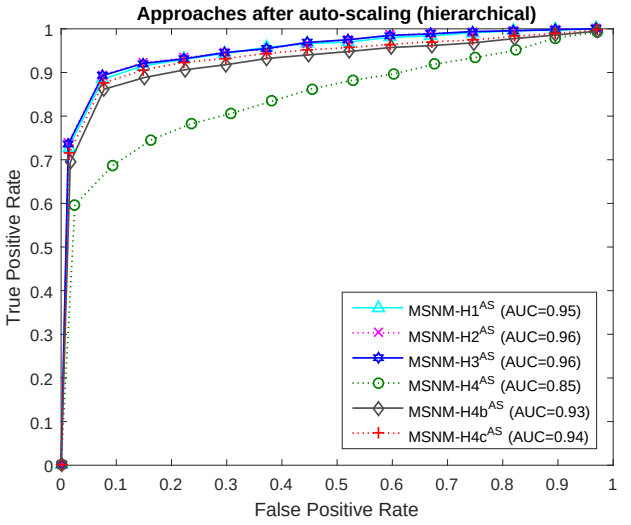
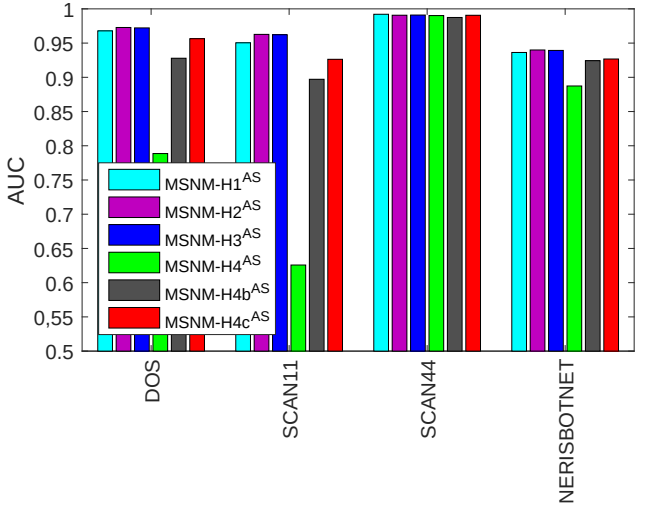


Fig. 8.14 Hierarchical topologies (a) **H4b** and (b) **H4c**.

Fig. 8.15 shows that both **H4b** and **H4c** have a detection performance similar to those obtained for the rest of the hierarchical scenarios. Results for **H4c** are somewhat better than for **H4b** due to the weighting. Unfortunately, applying solutions similar to **H4b** an **H4c** to the other scenarios (**H2** and **H3**) largely degrades performance in **H3**. This deserves further study.



(a)



(b)

Fig. 8.15 Comparison between hierarchical fusions, including H4b and H4c for (a) ROC curves and (b) AUC per individual attack.

Diagnosis (Step 4). After the detection step, the anomalous observations that correspond to the attacks in the test dataset are diagnosed. Table 8.6 show the most relevant features for these attacks after the diagnosis at the top layer of the hierarchy. These results are, in general terms, correct in terms of identifying the true location of the attack. `DoS` and `Scan11` attacks are properly fdetected in VR2. **H1** and **H2** only signal the `botnet` in VR2, which is an incomplete diagnosis (VR1 is also affected by the `botnet`). Yet, this is useful to prioritize the alarms: the operators should investigate (and probably solve) first VR2. Finally, it can observed that the diagnosis is also correct for the `scan44` attack. However, in **H1** only VR1 is signaled, although the attack takes place also in VR2.

Attack	H1	H2	H3	H4b	H4c
DoS	QVR2	QVR2	QVR12	QVR23, DVR23	QVR23, DVR23
Scan11	QVR2	QVR2	QVR12	QVR23, DVR23	DVR23, QVR23
Scan44	QVR1	QVR13	QVR12	QVR1, DVR23, QVR23	DVR23, QVR23, QVR1
Botnet	QVR2	QVR2	QVR12	QVR1, QVR23, DVR23	DVR23, QVR23, QVR1

Table 8.6 Diagnosis for the hierarchical fusion applying U-Squared at the top layer.

Next step is to propagate the diagnosis to low layers according to the diagnosis in the top layer. Table 8.7 shows the diagnosis for the second layer, while Table 8.8 represents the diagnosis for the ground layer. The single routers are diagnosed in the ground layer for the corresponding attacks. For the second layer, the diagnosis is not applied when the result in the top layer signals a leaf router (*e.g.* VR2) and not a branch (*e.g.* VR13), since this diagnosis corresponds to the ground layer. When the diagnosis is not performed, it is represented with a '-'. The leaf routers signaled are correct for all the attacks. Note also that **H3** prioritizes VR2 for the `botnet` attack, signaling the `IRC` features. Once the virtual routers are signaled, the diagnosis

is performed in the same way as we did before (for the standard fusion of the data), yielding congruent results with those obtained in such part.

Attack	H2	H3	H4b	H4c
DoS	-	sport_telnetVR2*	QVR2	QVR2
Scan11	-	sport_telnetVR2*, dport_kpasswdVR2, dport_gopherVR2	QVR2	QVR2
Scan44	-	QVR1	QVR2	QVR2
Botnet	-	dport_ircVR2, sport_ircVR2	QVR2	QVR2

Table 8.7 Diagnosis for the hierarchical fusion applying U-Squared at the second layer. When this diagnosis does not apply, it is signaled with '-'.

Attack	VR1	VR2	VR3
DoS	-	sport_telnetVR2*	-
Scan11	-	sport_telnetVR2*, dport_kpasswdVR2, dport_gopherVR2	-
Scan44	sport_kpasswdVR1, dport_kpasswdVR1, sport_quoteVR1, sport_snmpVR1, sport_discardVR1, dport_syslogVR2, sport_ftp_dataVR1	dport_kpasswdVR2, dportdport_gopherVR2, dport_emuleVR2, dport_msnmessengerVR2, sport_kpasswdVR1	-
Botnet	dport_ircVR1, sport_ircVR1	dport_ircVR2, sport_ircVR2	-

Table 8.8 Diagnosis for the leaf routers applying U-Squared. When this diagnosis does not apply, it is signaled with '-'. The '*' denotes that this is not considered an incorrect diagnosis, since we previously verified that, indeed there exists anomalous `telnet` traffic in the background.

The De-Parsing is omitted for this part of the evaluation, since there are no differences between standard and hierarchical MSNM for this step.

8.5 Comparison of Hierarchical and Standard Approaches

Fig. 8.16 compares the standard and the hierarchical fusions, showing that both types of union of the data have an equivalent performance. Applying the **C** and the **H*** fusions enables the identification of the source of the anomaly, as well as the consideration of the correlations between the data sources. However, the model is more complex for **C** than for **H***, which is more evident as the

number of data sources increases, being this the main inconvenience for the **C**-fusion (see Appendix D). In general, the hierarchical union of the data shows the following benefits:

- Maintaining the ***C**-fusion advantages*: identification of the source and/or location of the anomaly.
- *Volume and time consumption reduction* of the data needed for the monitoring. The hierarchical fusion yields a higher number of models with lower number of features than the standard fusion, which is more efficient in terms of resources used (see Appendix **??**).
- *Scalability*, since a higher number of sources can be added to the architecture of the hierarchy, yielding more possible scenarios.
- *Privacy*, since it is not needed to send the features to the integrator (that might be external to the organization). Instead, a high-level fusion can be performed so that sensitive data are not disclosed (*e.g.* sending the statistics to the integrator).

Thus, considering the good performance of **H*** models and their advantages, we believe these models deserve further research.

8.6 Conclusions

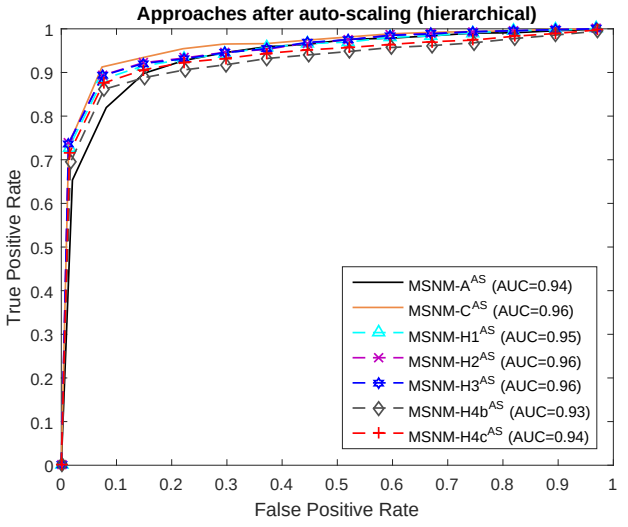
This chapter presents the evaluation of MSNM with real network data following two alternatives for the standard fusion of the data: the first alternative (**A**-fusion) aggregates the features of different sensors in a single matrix, reducing the dimensionality and simplifying the model building; the second alternative (**C**-fusion) concatenates the features collected from different sensors, enriching the model due to the consideration of the correlation of the variables between the sensors. Both **A**- and **C**-fusion present a high capability

of anomaly detection in the UGR'16 dataset, with an AUC higher than 0.9 in all the cases. The **C**-fusion alternative shows as a main benefit the capability of identifying the location of the anomaly, while for the **A**-fusion approach the greatest advantage is its technical performance related to volume and time consumption due to the lower number of features. However, the **A**-fusion alternative cannot distinguish the location of the anomaly.

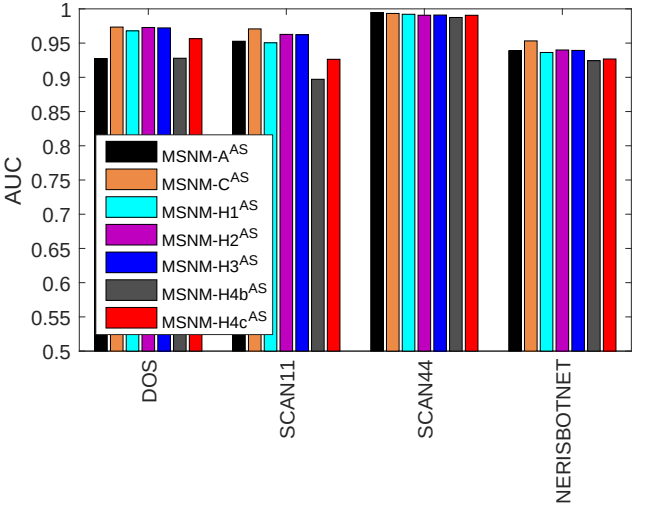
In the experiments shown in this chapter, the MSNM extensions for the pre-processing have a performance comparable with the AS. It still remains an open task to design and introduce more challenging attacks that might be undetected by some of the methods, so that we can assess the real performance of the detection methods more comprehensively.

The diagnosis helps to discover the causes of an anomaly. The experiments with real data have confirmed that U-Squared enhances the diagnosis, thus contributing to solve the *smearing* problem and reducing the complexity of the diagnosis. This also helps us to focus on the most relevant features to prioritize the search of the root causes of the anomaly.

Different scenarios are studied for the hierarchical fusion of the data. The first case of study has two layers: ground and top layer (**H1**); while the rest of the scenarios have three layers: ground, second and top layer (**H2** to **H4**). These scenarios are evaluated to study the effect of applying different organizations of the routers to the same architecture (**H2** to **H4**). The results are comparable to those obtained for the standard fusion of the data (in terms of the AUC), except for some cases that deserve further study.



(a)



(b)

Fig. 8.16 Comparison between standard and hierarchical fusions for (a) ROC curves and (b) AUC per individual attack.

Part IV

Conclusions

9
Conclusions

"Success is to love yourself, love what you do, and love how do you do."
Maya Angelou, American writer, poet, singer and civil rights activist

"We can only see little of the future, but enough to realize that there is much to do."
Alan Turing, Mathematician and Logician

Contents

9.1 Conclusions 220
9.2 Future Work 223

The core of this PhD is Multivariate Statistical Network Monitoring (MSNM), which is grounded on two pillars: on the one hand, from the IT Security arena, Network Security Monitoring (NSM); on the other hand, from the industrial processing arena, Multivariate Statistical Process Control (MSPC). Since MSNM was proposed almost five years ago, there have been a number of extensions of the methodology, either improving any of the steps [40, 129, 192, 195] or including new steps that add functionalities to the original proposal [33, 36]. These extensions can be used together, thus enhancing the performance of MSNM. This PhD work includes some of these extensions, such as [36, 40]. More precisely, the two main contributions presented in this work are also MSNM extensions:

- A pre-processing approach for the batch MSPC or cyclo-stationary MSNM that considers more observations to obtain the pre-processing parameters: *PARAmeters from More Observations (PARAMO)*. This approach improves the capability of anomaly detection of the monitoring system [192].

- A univariate diagnosis method to enhance the diagnosis: Univariate Squared (U-Squared). This method reduces the complexity of the diagnosis, improving the prioritization of the anomalies, which is one of the main problems for IT security teams. It also contributes to solve the *smearing* problem, which is well known in the context of the MSPC diagnosis.

Additionally, a methodology for the comparison of diagnosis methods is proposed. This methodology *i)* generates anomalies with known diagnosis, *ii)* defines a metric for the evaluation of the performance of the diagnosis methods, and *iii)* considers the factors affecting the diagnosis for the design of experiments. These requirements are applied following a Monte Carlo procedure, yielding low uncertainty results that, in combination with ANOVA, allow to compare the diagnosis methods in an accurate way.

Tables 9.1 and 9.2 summarize the objectives of this thesis. The goal of this PhD was to deal with the main research problems related to the detection and diagnosis of incidents in network security by applying multivariate data analysis. Table 9.1 shows the individual objectives that were defined to achieve this goal, while Table 9.2 presents some additional specific objectives also defined as a part of the research plan for this PhD.

Objective	Description
MO1	To design methods or algorithms for anomaly detection based in multivariate analysis. These methods should reduce the number of false alarms and allow the detection of *zero-day* attacks.
MO2	To design methods or algorithms for the accurate diagnosis of anomalies.

Table 9.1 Main objectives (MO).

The main conclusions of this PhD work have been detailed individually in the corresponding chapters. This chapter groups all these conclusions, providing the reader with a global vision about the results obtained during the research work. Finally, new lines of research derived from this thesis are proposed at the end of the chapter.

Objective	Description
SO1	To evaluate the proposed techniques for anomaly detection and diagnosis, which implies the comparison with the state-of-the-art methods.
SO2	To apply the proposed techniques to real network data.

Table 9.2 Secondary Objectives (SO).

9.1 Conclusions

This research work tackles a number of open points in cybersecurity, more precisely, in NSM:

1) The improvement in the capability of anomaly detection. The goal is to increase the sensitivity and the quality of a monitoring system to detect new anomalies without increasing the number of false positives.

2) The prioritization and interpretation of anomalies or events in a given time frame. This is also a way to mitigate the occurrence of false positives. The goal is to reduce the work load for security operators, and a visualization is worth a thousand words.

This work also deals with some relevant issues related to MSPC, contributing at the same time to solve some of the aforementioned open points in NSM:

3) The uncertainty in the pre-processing parameters [91, 92].

4) The enhancement of the capability of fault detection. The uncertainty may affect the quality of the monitoring system [91, 92].

5) The *smearing* effect in the diagnosis. This problem makes the diagnosis more complicated, since the variables signaled as anomalous might not be actually contaminated [114, 212].

In this context, PARAMO is presented as a pre-processing approach alternative to the reference method in the literature, Trajectory Centering and Scaling (TCS) [149]. This proposal contributes to solve points 3) and 4) in BMSPC, and also 1) in NSM; thus covering one of the main goals of this thesis (MO1) and one secondary objective (SO1). U-Squared is proposed as a univariate alternative for the diagnosis, tackling point 5) in MSPC, and 2) in NSM. This proposal covers the other main goal of the PhD (MO2), and also the other secondary objective (SO1).

- PARAMO uses more observations than TCS to obtain the pre-processing parameters, reducing the uncertainty of the pre-processing parameters.
- The reduction in the uncertainty makes the model to be more stable, which permits an increase of the capability for anomaly detection, reducing the number of non-detected faults. This enhances the quality of the monitoring system.
- U-Squared consists in applying observation-based Missing-data method for Exploratory Data Analysis (oMEDA) [29] in the full variable space, instead of individually to the model and the residual sub-spaces.
- The U-Squared is a univariate method that enhances the diagnosis. One of the reason is that it eliminates the *smearing* effect, since it does not consider the correlations of the variables. This simplifies the diagnosis process and helps the prioritization of events.

PARAMO is assessed with different data sets, which were generated by simulating the *Saccaromyces Cerevisiae* process cultivation. U-Squared is evaluated with synthetic data simulated for different correlation levels, following the proposed methodology in this thesis for comparing diagnosis methods [195]. The results for U-Squared are also validated with two independent data sets corresponding to *i)* a virtual communications network and *ii)* the simulation of the *Saccaromyces Cerevisiae* process cultivation.

Both PARAMO and U-Squared are also validated with real network data, thus achieving the secondary objective SO2. The UGR'16 dataset [130] is used in this evaluation. MSNM is applied over two alternatives for the standard organization of the data: *i)* by concatenating the features of the different data sources in a single matrix (**C**-fusion), and *ii)* by aggregating all the extracted features from different data sources in a single matrix (**A**-fusion). The goal was to study the effect of applying PARAMO as a pre-processing approach on these two different alternatives, since PARAMO takes into account the cyclo-stationarity of the data. Another goal of this evaluation was to compare U-Squared with a multivariate method (oMEDA) for the diagnosis. The conclusions derived from this part are:

- **A**-fusion allows simplifying the fusion but not identifying the location of the anomaly. On the contrary, **C**-fusion allows to distinguish the location of the anomaly but it needs more technical resources (*e.g.* computational time) than the **A**-fusion due to the high number of features considered in this type of fusion. **C**-fusion also reduces model stability. **C**-fusion can be applied to combine different types of data sources while **A**-fusion is restricted to the same type.

- eXponential PARAMO (X-PARAMO) provides similar results to Auto-Scaling (AS). The main advantage of X-PARAMO over AS is that X-PARAMO show better performance to detect anomalies affecting the cyclo-stationarity of the data.

- U-Squared enhances the diagnosis, reducing the complexity of the diagnosis and helping the prioritization in the search of the root causes of an anomaly. This allows to perform an easy and quick diagnosis.

In this research work, the hierarchical fusion for the MSNM methodology [129] is evaluated for the first time with real network data, using the UGR'16 dataset [130] (**SO2**). Four hierarchical scenarios are assessed to

study the effect of the different organizations of the data in this type of fusion. The hierarchical approach shows the following benefits over the standard approaches: ***C**-fusion advantages* are maintained (identification of the location of the anomaly), *volume and time consumption reduction* of the data needed for the monitoring, *scalable architecture*, and *privacy increasing*.

The diagnosis for a hierarchical fusion starts at the top level, helping operators to locate the anomaly. At the ground level (and sometimes in the intermediate levels), the diagnosis helps to discover the causes of an anomaly by identifying the most relevant features involved in such anomaly. In this study, the U-Squared is applied to enhance the diagnosis.

In summary, this PhD provides a general insight in the MSNM methodology: from the basis in cybersecurity and industrial process control to the most recent extensions for MSNM presented as main contributions of the thesis. The proposals are evaluated both with simulated data and real network data. In addition, these proposals are also validated for the two fields of application: network security monitoring and industrial process control. Although the hierarchical organization is not a contribution of this thesis *per se*, it is applied for the first time to real network data. In addition, its evaluation under different scenarios allows to identify the correlation problems that may appear depending on the distribution of the sensors.

9.2 Future Work

Applying PARAMO to the pre-processing provides similar performance to auto-scaling on real network data. The benefits of its application in fault detection when the faults affect the cyclo-stationarity of chemical data are also shown. Since all the methods provide similar performance in the real network data for the existing attacks, we hypothesize that the evaluation of more complex attacks may show a superior performance of 3-way pre-processing approaches. To evaluate the real profit of PARAMO in real network data,

it is needed to design and introduce new attacks breaking the normal time-model. Take as an example, sending SPAM e-mails in a low rate for several consecutive days at unusual times of the day. This behavior usually might not generate anomalous events for auto-scaled models. However, pre-processing with PARAMO may allow detecting small deviations of the data from the average of each specific time of the day. Thus, the number of true positives can be increased for this type of attacks, at the same time that the rest of attacks is still detected. We leave as future research work the tasks of designing and introducing attacks affecting the normal time-model in the UGR'16, as well as evaluating the effect of applying PARAMO when there exist such attacks.

Designing a good hierarchical scheme needs to take into account the existing correlations in a proper manner. The organization of the sensors may affect the correlation of the data and, thus, the capability of anomaly detection of the monitoring system. A future research work is to study the optimal distribution to define the hierarchical schemes and the number of layers used to build the hierarchy. In addition, one of the alternatives proposed to deal with the problems derived of the distribution of the sensors is to weight each branch in the hierarchy tree. This allows to balance the importance of each block in the model, which is more relevant when we combine features obtained from the raw data and statistics already computed in another layer in the hierarchy. Another possibility is applying high-level fusion in all the layers of the hierarchy and also weighting the branches. The way of selecting the weights, as well as to study their effect in different hierarchical scenarios is also a new line of research.

Finally, there have been extensions of MSNM for each of the steps except for the parsing (Step 1). Thus, the proposal of alternatives for the parsing step of the MSNM methodology still remains an open problem.

References

[1] A., W. J., Theodora, K., and F., M. J. (1999). Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13(304):397–413.

[2] Afanador, N. L., Smolinska, A., Tran, T. N., and Blanchet, L. (2016). Unsupervised random forest: a tutorial with case studies. *Journal of Chemometrics*, 30:232–241.

[3] Aiello, M., Mongelli, M., Cambiaso, E., and Papaleo, G. (2016). Profiling DNS tunneling attacks with PCA and mutual information. *Logic Journal of IGPL*, 24(6):957–970.

[4] Alcala, C. F. and Joe Qin, S. (2011). Analysis and generalization of fault diagnosis methods for process monitoring. *Journal of Process Control*, 21(3):322–330.

[5] Alcala, C. F. and Qin, S. J. (2009). Reconstruction-based contribution for process monitoring. *Automatica*, 45(7):1593–1600.

[6] AlienVault (2019). Compare AlienVault Products. https://www.alienvault.com/products/ossim/compare. [Online; accessed 08-Nov-2019].

[7] Alliance, C. T. (2019). Cyber Threat Alliance. https://www.cyberthreatalliance.org/. [on-line visited on 20/10/2019].

[8] Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2nd edition.

[9] Alpcan, T. and Basar, T. (2011). *Network Security. A Decision and Game-Theoretic Approach*. Cambridge University Press.

[10] Alves, G. (2018). Discovering som, an unsupervised neural network. https://medium.com/neuronio/discovering-som-an-unsupervised-neural-network-12e787f38f9.

[11] AT&T-cybersecurity (2012). Open Threat Exchange (OTX). https://www.alienvault.com/open-threat-exchange. [Online; accessed 08-Nov-2019].

[12] AT&T-cybersecurity (2016). AlienVault® OSSIM™, Open Source Security Information and Event Management (SIEM). https://www.alienvault.com/products/ossim. [Online; accessed 08-Nov-2019].

[13] AT&T-cybersecurity (2019). AlienVault® Unified Security Management® (USM). https://www.alienvault.com/products. [Online; accessed 08-Nov-2019].

[14] AT&T-Cybersecurity (2019). The Art of Triage: Types of Security Incidents. https://www.alienvault.com/resource-center/ebook/insider-guide-to-incident-response/types-of-security-incidents. [Online; accessed 08-Nov-2019].

[15] AVAST (2015). Malware and Antimalware. https://www.avast.com/es-es/c-malware.

[16] Bace, R. G. (2000). *Intrusion Detection*. Macmillan Technical Publishing (Technology Series).

[17] Badr, W. (2019). Auto-Encoder: What Is It? And What Is It Used For? Technical report, Towards Data Science, https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726. [Online; accessed 18-Oct-2019].

[18] BBC News (2019). Facebook to be fined 5bn Dollars over Cambridge Analytica scandal. https://www.bbc.com/news/world-us-canada-48972327. [Online; accessed 31-Oct-2019].

[19] Bejtlich, R. (2005). *The TAO of the Network Security Monitoring. Beyond Intrusion Detection*. Addison-Wesley.

[20] Bejtlich, R. (2013). *The practice of Network Security Monitoring*. No Starch Press.

[21] Bikfalvi, A. (2012). Advanced box plot for matlab. http://alex.bikfalvi.com/research/advanced_matlab_boxplot. [Online; accessed 15-Mar-2016].

[22] Box, G. E. P. (1954). Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems: Effect of Inequality of Variance in One-Way Classification. *The Annals of Mathematical Statistics*, 25:290–302.

[23] Brereton, R. (2009). *Chemometrics for Pattern Recognition*, chapter Exploratory Data Analysis, pages 47–106. John Wiley & Sons.

[24] Bullard, C. (2014). Argus. Technical report, QoSient, https://openargus.org/documentation. [Online; accessed 05-Sep-2019].

[25] Cadwalladr, C. and Graham-Harrison, E. (2018). Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election. [Online; accessed 08-Nov-2019].

[26] Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., and Pepe, T. (2011). A novel PCA-based network anomaly detection. In *IEEE International Conference on Communications*.

[27] Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., and Pepe, T. (2014). Improving PCA-based anomaly detection by using multiple time scale analysis and Kullback-Leibler divergence. *International Journal of Communication Systems*, 27(10):1731–1751.

[28] Camacho, J. (2010). Missing-data theory in the context of exploratory data analysis. *Chemometrics and Intelligent Laboratory Systems*, 103(1):8–18.

[29] Camacho, J. (2011). Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models. *Journal of Chemometrics*, 25(11):592–600.

[30] Camacho, J. (2014). Visualizing Big data with Compressed Score Plots: Approach and research challenges. *Chemometrics and Intelligent Laboratory Systems*, 135:110 – 125.

[31] Camacho, J. (2016). On the Generation of Random Multivariate Data. *Chemometrics and Intelligent Laboratory Systems*, 160:40–51.

[32] Camacho, J. (2019). Gestión de Incidentes de Seguridad. Máster Propio en Ciberseguridad de la Universidad de Granada (material no publicado).

[33] Camacho, J., Bro, R., and Kotz, D. (2019a). Networkmetrics unraveled: MBDA in Action. *Submitted to CoRR (arXiv:1907.02677v1 [cs.NI])*.

[34] Camacho, J. and Ferrer, A. (2012). Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects. *Journal of Chemometrics*, 26(7):361–373.

[35] Camacho, J. and Ferrer, A. (2014). Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: Practical aspects. *Chemometrics and Intelligent Laboratory Systems*, 131:37–50.

[36] Camacho, J., García-Giménez, J. M., **Fuentes-García, N. M.**, and Maciá-Fernández, G. (2019b). Multivariate Big Data Analysis for Intrusion Detection: 5 steps from the haystack to the needle. *Computers and Security (COSE)*, 87.

[37] Camacho, J. and García-Jiménez, J. M. (2018). FCParser. https://github.com/josecamachop/FCParser.

[38] Camacho, J., García-Teodoro, P., and Maciá-Fernández, G. (2017a). Traffic Monitoring and Diagnosis with Multivariate Statistical Network Monitoring: A Case Study. *IEEE Security & Privacy International Workshop on Traffic Measurements for Cybersecurity (WTMC 2017)*, pages 241–246.

[39] Camacho, J., Lauri, D., Lennox, B., Escabias, M., and Valderrama, M. (2015a). Evaluation of smoothing techniques in the run to run optimization of fed-batch processes with u-PLS. *Journal of Chemometrics*, 29(6):338–348.

[40] Camacho, J., Maciá-Fernández, G., **Fuentes-García, N. M.**, and Saccenti, E. (2017b). Semi-supervised multivariate statistical network monitoring for learning security threats. *Transactions on Information Forensics and Security*, 14(8):2179–2189.

[41] Camacho, J., Maciá-Fernández, G., Verdejo, J. E. D., and García-Teodoro, P. (2014). Tackling the Big Data 4 Vs for Anomaly Detection. *INFOCOM'2014 Workshop on Security and Privacy in Big Data*, pages 500–505.

[42] Camacho, J., Pérez-Villegas, A., García-Teodoro, P., and Maciá-Fernández, G. (2016). PCA-based multivariate statistical network monitoring for anomaly detection. *Computers & Security*, 59:118–137.

[43] Camacho, J., Pérez-Villegas, A., Rodríguez-Gómez, R. A., and Jiménez-Mañas, E. (2015b). Multivariate Exploratory Data Analysis (MEDA) Toolbox for Matlab. *Chemometrics and Intelligent Laboratory Systems*, 143:49–57.

[44] Camacho, J. and Pico, J. (2006). Multi-phase principal component analysis for batch processes modelling. *Chemometrics and Intelligent Laboratory Systems*, 81(2):127–136.

[45] Camacho, J., Picó, J., and Ferrer, A. (2007). Self-tuning run to run optimization of fed-batch processes using unfold-PLS. *AIChE Journal*, 53(7):1789–1804.

[46] Camacho, J., Picó, J., and Ferrer, A. (2008a). Bilinear modelling of batch processes. Part I: Theoretical discussion. *Journal of Chemometrics*, 22(5):299–308.

[47] Camacho, J., Picó, J., and Ferrer, A. (2008b). Bilinear modelling of batch processes. Part II: A comparison of PLS soft-sensors. *Journal of Chemometrics*, 22(10):533–547.

[48] Camacho, J., Picó, J., and Ferrer, A. (2008c). Multi-phase analysis framework for handling batch process data. *Journal of Chemometrics*, 22(11-12):632–643.

[49] Camacho, J., Pico, J., and Ferrer, A. (2009). On-line monitoring of batch processes based on PCA: Does the modelling structure matter? *Analytica Chimica Acta*, 642:59–68.

[50] Camacho, J., Rodríguez-Gómez, R., and Saccenti, E. (2017c). Group-wise Principal Component Analysis for Exploratory Data Analysis. *Journal of Computational and Graphical Statistics*, 26(3):501–512.

[51] Camacho, J. and Rodríguez-Gómez, R. A. (2015). MEDA Toolbox. https://github.com/josecamachop/MEDA-Toolbox. [Latest update to stable version performed in 2018].

[52] Camacho-Páez, J. (2007). *New Methods Based on the Projection to Latent Structures for Monitoring, Prediction and Optimization of Batch Processes*. phdthesis, Technical University of Valencia.

[53] Chen, J. and Liu, K. (2002). On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chemical Engineering Science*, 57(1):63–75.

[54] Chen, Z., Yeo, C. K., Lee, B. S., and Lau, C. T. (2016). Detection of Network Anomalies using Improved-MSPCA with Sketches. *Computers & Security*, 65:314–328.

[55] Cherkassky, V. and Mulier, F. (2007). *Learning from Data: Concepts, Theory, and Methods*. Wiley-IEEE Press, 2nd edition.

[56] Chiang, L. H. and Braatz, R. D. (2003). Process monitoring using causal map and multivariate statistics: Fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(142):159–178.

[57] Chiang, L. H., Jiang, B., Zhu, X., Huang, D., and Braatz, R. D. (2015). Diagnosis of multiple and unknown faults using the causal map and multivariate statistics. *Journal of Process Control*, 28(142):27–39.

[58] Choksi, K., Shah, B., and Kale, O. (2014). Intrusion Detection System using Self Organizing Map: A Survey. *International Journal of Engineering Research and Applications*.

[59] Cisco and Sourcefire (1998). Snort. https://www.snort.org/. [Online; accessed 01-Sep-2019].

[60] Cisco-Systems (2004). Cisco Systems NetFlow Services Export Version 9. Technical report, Cisco Systems, https://tools.ietf.org/html/rfc3954. [Online; accessed 01-Sep-2019].

[61] Cocchi, M. (2019). *Data Fusion Methodology and Applications*, volume 31, chapter Introduction: Ways and Means to Deal With Data From Multiple Sources, pages 1–26. Elsevier.

[62] Collins, M. (2014). *Network Security Through Data Analysis. Building situational awareness*. O'Reilly.

[63] Combs, G. (1998). Wireshark. https://www.wireshark.org/. [Online; accessed 01-Sep-2019].

[64] CSIRT-Gadgets (2019). The FASTEST Way to Consume Threat Intelligence. https://csirtgadgets.com/commits/2018/1/6/the-fastest-way-to-consume-threat-intel. [Online; accessed 08-Nov-2019].

[65] Dayal, B. and MacGregor, J. (1997). Recursive exponentially weightes pls and its applications to adaptive control and prediction. *Journal of Process Control*, 7:169–179.

[66] Delimargas, A., Skevakis, E., Halabian, H., and Lambadaris, I. (2014). Evaluating a modified PCA approach on network anomaly detection. *Fifth International Conference on Next Generation Networks and Services (NGNS)*, pages 124–131.

[67] Doeswijk, T. G., Smilde, A. K., Hageman, J. A., Westerhuis, J. A., and van Eeuwijk, F. A. (2011). On the increase of predictive performance with high-level data fusion. *Analytica Chimica Acta*, 705:41–47.

[68] Dua, S. and Du, X. (2016). *Data mining and machine learning in cybersecurity*. CRC press.

[69] Dunia, R. and Joe Qin, S. (1998). Subspace approach to multidimensional fault identification and reconstruction. *AIChE Journal*, 44(8):1813–1831.

[70] Elastic (2000). X-pack. https://www.elastic.co/es/what-is/open-x-pack. [Online; accessed 01-Sep-2019].

[71] Elasticsearch (2019). The Elastic Stack. Meet the core products. https://www.elastic.co/es/products/elastic-stack. [Online; accessed 01-Sep-2019].

[72] Eleven Paths (2016). Computer Emergency Response Team (CERT) with Leonardo Huertas. Technical report, Eleven Paths, https://www.elevenpaths.com/es/noticias-y-eventos/elevenpaths-talks/equipo-de-respuesta-ante-emergencias-informaticas-cert-del-ingles-computer-emergency-response-team/index.html. [Online; accessed 15-Sep-2019].

[73] ENISA (2019). CSIRTs by Country - Interactive Map. https://www.enisa.europa.eu/topics/csirts-in-europe/csirt-inventory/certs-by-country-interactive-map. [Online; accessed 15-Sep-2019].

[74] Fernandes, G., Carvalho, L. F., Rodrigues, J. J., and Proença, M. L. (2016). Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization. *Journal of Network and Computer Applications*, 64:1–11.

[75] Ferrer, A. (2007). Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process. *Quality Engineering*, 19:311–325.

[76] Ferrer, A. (2014). Latent structures-based multivariate statistical process control: A paradigm shift. *Quality Engineering*, 26(1):72–91.

[77] Finkle, J. (2018). Twitter urges all users to change passwords after glitch. https://www.reuters.com/article/us-twitter-passwords/twitter-urges-all-users-to-change-passwords-after-glitch-idUSKBN1I42JG. [Online; accessed 18-Oct-2019].

[78] Fung, B. (2019). Facebook will pay an unprecedented 5 billion Dollars penalty over privacy breaches. https://edition.cnn.com/2019/07/24/tech/facebook-ftc-settlement/index.html.

[79] García, S., Grill, M., Stiborek, J., and Zunino, A. (2014a). An empirical comparison of botnet detection methods. *Computers and Security Journal*, 45:100–123.

[80] García, S., Grill, M., Stiborek, J., and Zunino, A. (2014b). The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. https://www.stratosphereips.org/datasets-ctu13.

[81] García-Teodoro, P., Díaz-Verdejo, J., and Maciá-Fernández, E. V. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1):18–28.

[82] García-Teodoro, P., Díaz-Verdejo, J. E., and López-Soler, J. M. (2014). *Transmisión de datos y redes de computadores*. Pearson.

[83] Gartner (2018). Gartner Forecasts Worldwide Information Security Spending to Exceed 124 Billion Dollars in 2019. https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019. [Online; accessed 01-Ago-2019].

[84] Gartner (2019a). Gartner Identifies the Top Seven Security and Risk Management Trends for 2019. https://www.gartner.com/en/newsroom/press-releases/2019-03-05-gartner-identifies-the-top-seven-security-and-risk-ma. [Online; accessed 01-Ago-2019].

[85] Gartner (2019b). Gartner Magic Quadrant. https://www.gartner.com/en/research/methodologies/magic-quadrants-research. [Online; accessed 08-Nov-2019].

[86] Gartner (2019c). Unified Threat Management (utm). https://www.gartner.com/en/information-technology/glossary/unified-threat-management-utm. [Online; accessed 17-Sep-2019].

[87] Gartner (2019d). What is Security Information and Event Management (SIEM)? https://www.gartner.com/reviews/market/security-information-event-management. [Online; accessed 17-Sep-2019].

[88] Golub, G. and Van Loan, C. (1996). *Matrix Computations*. University Press, Baltimore, MD, USA (1996), 3rd edition.

[89] González, J. M., Camacho, J., and Ferrer, A. (2018). MVBatch: A matlab toolbox for batch process modeling and monitoring. *Chemometrics and Intelligent Laboratory Systems*, 183:122–133.

[90] González-Martínez, J., Ferrer, A., and Westerhuis, J. (2011). Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping. *Chemometrics and Intelligent Laboratory Systems*, 105:195–206.

[91] González-Martínez, J. M. (2015). *Advances on bilinear modeling of biochemical batch processes*. PhD thesis, Universitat Politècnica de València.

[92] González-Martínez, J. M., Camacho, J., and Ferrer, A. (2013). Bilinear modelling of batch processes. Part III: parameter stability. *Journal of Chemometrics*, 28:10–27.

[93] González-Martínez, J. M., Noord, O. E., and Ferrer, A. (2014a). Multisynchro: a novel approach for batch synchronization in scenarios of multiple asynchronisms. *Journal of Chemometrics*, 28(5):462–475.

[94] González-Martínez, J. M., **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2017). Parameter stability and its effects on bilinear modelling of batch processes. In *Mini Arctic Workshop, Valencia (Spain)*.

[95] González-Martínez, J. M., Vitale, R., de Noord O. E., and Ferrer, A. (2014b). Effect of Synchronization on Bilinear Batch Process Modeling. *Industrial & Engineering Chemistry Research*, 53(53):4339–4351.

[96] Grothaus, M. (2019). The phone numbers of 419 million Facebook accounts have been leaked. https://www.fastcompany.com/90399734/the-phone-numbers-of-419-million-facebook-accounts-have-been-leaked. Accessed online on 12/10/19.

[97] Hanley, J. and McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

[98] Hartwig, F. and Dearing, B. E. (1979). *Exploratory Data Analysis*. Sage University Paper, 8th printing edition.

[99] Hernández-Moreno, A. (2019). Formal start of the conference JNIC 2019. In *V Jornadas Nacionales de Investigación en Ciberseguridad*.

[100] Hubens, N. (2018). Deep inside: Autoencoders. Technical report, Towards Data Science, https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f. [Online; accessed 08-Nov-2019].

[101] ICANN (Internet Corporation for Assigned Names and Numbers) (1997). Whois. https://lookup.icann.org/. [Online; accessed 03-Sep-2019].

[102] Inc., G. (2018). Gartner 2018 Magic Quadrant for SIEM. https://www.splunk.com/en_us/form/gartner-siem-magic-quadrant.html?utm_campaign=google_emea_tier2_en_search_generic_security_siem&utm [Online; accessed 18-Oct-2019].

[103] INCIBE (2017). Diseño y Configuración de IPS, IDS y SIEM en Sistemas de Control Industrial. https://www.incibe-cert.es/blog/diseno-y-configuracion-ips-ids-y-siem-sistemas-control-industrial. [Online; accessed 03-Sep-2019].

[104] INCIBE (2018). Respuesta a incidentes. politicas de seguridad para la pyme. https://www.incibe.es/sites/default/files/contenidos/politicas/documentos/respuesta-incidentes.pdf. [Online; accessed 03-Sep-2019].

[105] Iturbe, M. (2017). *Data-Driven Anomaly Detection in Industrial Networks*. PhD thesis, Mondragon Unibertsitatea.

[106] J, M. and Kourti (1995). Statistical Process Control of Multivariate Processes. *Control Engineering Practice*, 3(3):403–414.

[107] Jackson, J. E. and Mudholkar, G. S. (1979). Control procedures for residuals associated with Principal Component Analysis. *Technometrics*, 21:331–349.

[108] Jacobson, V., Leres, C., and McCanne, S. (1988). tcpdump and libpcap. https://www.tcpdump.org/. [Online; accessed 03-Sep-2019].

[109] Jiang, D., Yao, C., Xu, Z., and Qin, W. (2015). Multi-scale anomaly detection for high-speed network traffic. *Transactions on Emerging Telecommunications Technologies*, 26(3):308–317.

[110] Kanaoka, A. and Okamoto, E. (2003). Multivariate statistical analysis of network traffic for intrusion detection. *14th international workshop on Database and Expert System Applications*, pages 1–5.

[111] Kaushik, S. (2016). An Introduction to Clustering and different methods of clustering. https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/.

[112] Kavanagh, K., Bussa, T., and Sadowski, G. (2018). Magic Quadrant for Security Information and Event Management. techreport, Gartner, https://www.gartner.com/en/documents/3894573/magic-quadrant-for-security-information-and-event-manage. [Online; accessed 15-Sep-2019].

[113] Kerkhof, P. V. D., Vanlaer, J., Gins, G., and Impe, J. F. M. V. (2013a). Analysis of smearing-out in contribution plot based fault isolation for Statistical Process Control. *Chemical Engineering Science*, pages 285–293.

[114] Kerkhof, P. V. D., Vanlaer, J., Gins, G., and Impe, J. F. M. V. (2013b). Contribution plots for Statistical Process Control : analysis of the smearing-out effect. *European Control Conference (ECC)*, pages 1–6.

[115] Kourti, T. (2003a). Abnormal situation detection, three-way data and projection methods; robust data archiving and modeling for industrial applications. *Annual Reviews in Control*, 27:131–139.

[116] Kourti, T. (2003b). Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17(1):93–109.

[117] Kourti, T. and MacGregor, J. F. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28(4):409–428.

[118] Lakhina, A., Crovella, M., and Diot, C. (2004). Diagnosing network-wide traffic anomalies. *ACM SIGCOMM Computer Communication Review*, 34(4):219.

[119] Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. *ACM SIGCOMM Computer Communication Review*, 35(4):217.

[120] Lei, F., Rotboll, M., and Jorgensen, S. B. (2001). A biochemically structured model for Saccharomyces cerevisiae. *Journal of Biotechnology*, 88(3):205–221.

[121] Lennox, B., Montague, G., Hiden, H., Kornfeld, G., and Goulding, P. (2001). Process monitoring of an industrial fed-batch fermentation. *Biotechnology and Bioengineering*, 74(2):125–135.

[122] Li, G., Alcala, C. F., Qin, S. J., and Zhou, D. (2011). Generalized reconstruction-based contributions for output-relevant fault diagnosis with application to the Tennessee Eastman process. *IEEE Transactions on Control Systems Technology*, 19(5):1114–1127.

[123] Li, G., Qin, S. J., and Chai, T. (2014). Multi-directional reconstruction based contributions for root-cause diagnosis of dynamic processes. *Proceedings of the American Control Conference*, pages 3500–3505.

[124] Litan, A. and Nicolett, M. (2014). Market Guide for User Behavior Analytics. https://www.gartner.com/en/documents/2831117/market-guide-for-user-behavior-analytics. [Online; accessed 17-Sep-2019].

[125] LogRhythm (2019). LogRhythm NextGen SIEM Platform. https://es.logrhythm.com/products/nextgen-siem-platform/. [Online; accessed 03-Sep-2019].

[126] Lonvick, C. (2001). The BSD syslog Protocol. Technical report, Cisco System, https://www.ietf.org/rfc/rfc3164.txt. [Online; accessed 27-Nov-2019].

[127] Lyon, G. (1997). Nmap (Network Mapper). https://nmap.org/. [Online; accessed 17-Ago-2019].

[128] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., **Fuentes-García, N. M.**, García-Teodoro, P., and Therón Sánchez, R. (2018). Un resumen de: UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red basados en cicloestacionariedad. In *IV Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), San Sebastián (Spain)*, pages 117–118.

[129] Maciá-Fernández, G., Camacho, J., García-Teodoro, P., and Rodríguez-Gómez, R. A. (2016). Hierarchical PCA-Based Multivariate Statistical Network Monitoring for Anomaly Detection. *International Workshop on Information Forensics and Security*.

[130] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., and Therón Sánchez, R. (2018). UGR'16: a new dataset for the evaluation of cyclostationarity-based network IDSs. *Computers & Security*, 73:411–424.

[131] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., **Fuentes-García, N. M.**, García-Teodoro, P., and Therón-Sánchez, R. (2017). UGR'16: Un nuevo conjunto de datos para la evaluación de IDS de red. In *XIII Jornadas de Ingenieria Telematica (JITEL2017), Valencia (Spain)*, pages 71–78.

[132] Magán-Carrión, R., Camacho, J., Maciá-Fernández, G., and **Fuentes-García, N. M.** (2017). Esquema Jerárquico de Monitorización y Detección de Anomalías en Red: Aplicación Práctica. In *III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), Madrid (Spain)*.

[133] Malwarebytes (2019). Malware. https://es.malwarebytes.com/malware/.

[134] Marchette, D. J. (2001). *Computer Intrusion Detection and Network Monitoring. A Statistical Viewpoint*. Springer (Statistics for Engineering and Information Science).

[135] Marty, R. (2010). *Applied Security Visualization*. Addison-Wesley.

[136] McCandless, D., Evans, T., Barton, P., Tomasevic, S., and Geere, D. (2019). Information Is Beautiful. World's Biggest Data Breaches & Hacks. https://informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/. [Online; accessed 08-Sep-2019].

[137] Mendes, R. and Vilela, J. P. (2017). Privacy-Preserving Data Mining: Methods, Metrics, and Applications. *IEEE Access*, 5:10562 – 10582.

[138] Metz, C. E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298.

[139] Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *Network and Distributed Systems Security Symposium, NDSS (arXiv:1802.09089v2)*.

[140] MITRE (1999). Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org/. [Online; accessed 03-Sep-2019].

[141] Molina, J. (2016). Threat Intelligence: el porqué de las cosas. https://www.welivesecurity.com/la-es/2016/12/01/threat-intelligence/. [Online; accessed 18-Oct-2019].

[142] Montgomery, D. C. (2017). *Design and analysis of experiments*. New York: John Wiley, 9 edition.

[143] MxToolbox, I. (2019). MxToolbox. https://mxtoolbox.com/. [Online; accessed 13-Ago-2019].

[144] National Institute of Standards and Technology (NIST) (2019). National Vulnerability Database (NVD). https://nvd.nist.gov/. [Online; accessed 14-Ago-2019].

[145] Networks, G. (2009). Open Vulnerability Assessment Scanner (OpenVAS). http://openvas.org/. [Online; accessed 18-Sep-2019].

[146] NG, C. (2019). Data Privacy: Definition, Explanation and Guide. Technical report, Varonis, https://www.varonis.com/blog/data-privacy/. [Online; accessed 17-Ago-2019].

[147] Nguyen, Q. P., Lim, K. W., Divakaran, D. M., Low, K. H., and Chan, M. C. (2019). GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection. In *2019 IEEE Conference on Communications and Network Security (CNS)*.

[148] Nomikos, P. and MacGregor, J. (1994). Monitoring batch processes using multiway principal components analysis. *AIChE Journal*, 40(8):1361–1375.

[149] Nomikos, P. and MacGregor, J. (1995a). Multivariate SPC Charts for Monitoring Batch Processes. *Technometrics*, 37(1):41–59.

[150] Nomikos, P. and MacGregor, J. F. (1995b). Multivariate Statistical Process Control Charts for Monitoring Batch Processes. *Technometrics*, 37(1):41–59.

[151] NoVirusThanks (2010a). IPVoid. https://www.ipvoid.com/. [Online; accessed 18-Ago-2019].

[152] NoVirusThanks (2010b). URLVoid. https://www.urlvoid.com/. [Online; accessed 18-Ago-2019].

[153] Open-Source (2019). NFDUMP. http://nfdump.sourceforge.net/. [Online; accessed 17-Ago-2019].

[154] OSSEC Project Team (2008). Open Source HIDS SECurity. https://www.ossec.net/. [Online; accessed 17-Ago-2019].

[155] Pan, Y. C., Dong, Y., and Qin, S. J. (2015). Fault Diagnosis Using Concurrent Projection to Latent Structures. *IFAC-PapersOnLine*, 48(8):1276–1281.

[156] Paxson, V. and Sommer, R. (1994). The Zeek Network Security Monitor (Bro). https://www.zeek.org/. [Online; accessed 18-Ago-2019].

[157] Polyakov, A. (2018). Machine Learning for Cybersecurity 101. Technical report, Towards Data Science. Sharing concepts, ideas, and codes, https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b. [Online; accessed 10-Oct-2019].

[158] Qin, S. J. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502.

[159] QoSient (2007). Ra Client. Technical report, https://manpages.debian.org/testing/argus-client/ra.1.en.html. [Online; accessed 15-Ago-2019].

[160] Ralhan, A. (2018). Self Organizing Maps. https://towardsdatascience.com/self-organizing-maps-ff5853a118d4.

[161] Ramaker, H., van Sprang, E., Westerhuis, J., Gurden, S., Smilde, A., and van der Meulen, F. (2006). Performance assessment and improvement of control charts for statistical batch process monitoring. *Statistica Neerlandica*, 60(3):339–360.

[162] Ramaker, H., van Sprang, E., Westerhuis, J., and Smilde, A. (2005). Fault detection properties of global, local and time evolving models for batch process monitoring. *Journal of Process Control*, 15(7):799–805.

[163] Rannar, S., MacGregor, J. F., and Wold, S. (1998). Adaptive batch monitoring using hierarchical PCA. *Chemometrics and Intelligent Laboratory Systems*, 41(1):73–81.

[164] Rasmus Bro and Age K. Smilde (2003). Centering and scaling in component analysis. *Journal of Chemometrics*, 17(1):16–33.

[165] Rato, T. J. and Reis, M. S. (2015a). On-line process monitoring using local measures of association. Part I: Detection performance. *Chemometrics and Intelligent Laboratory Systems*, 142(142):255–264.

[166] Rato, T. J. and Reis, M. S. (2015b). On-line process monitoring using local measures of association. Part II: Design issues and fault diagnosis. *Chemometrics and Intelligent Laboratory Systems*, 142(142):265–275.

[167] Researchers, U. (2011). Privacy vs. Confidentiality: What is the Difference? Technical report, University of California, Irvine (UCI), https://research.uci.edu/compliance/human-research-protections/docs/privacy-confidentiality-hrp.pdf. [Online; accessed 16-Sep-2019].

[168] Ringberg, H., Soule, A., Rexford, J., and Diot, C. (2007). Sensitivity of PCA for traffic anomaly detection. *ACM SIGMETRICS Performance Evaluation Review*, 35(1):109.

[169] Ruefle, R. (2007). Defining Computer Security Incident Response Teams. Technical report, CISA, https://www.us-cert.gov/bsi/articles/best-practices/incident-management/defining-computer-security-incident-response-teams. [Online; accessed 20-Ago-2019].

[170] Salah, S., Maciá-Fernández, G., and Díaz-Verdejo, J. E. (2013). A model-based survey of alert correlation techniques. *Computer Networks*, 57:1289–1317.

[171] Salah, S., Maciá-Fernández, G., and Díaz-Verdejo, J. E. (2018). Fusing Information from Tickets and Alerts to Improve the Incident Resolution Process. *Information Fusion*, 45:38–52.

[172] Sánchez-San-Venancio, M. Á. (2019). Formal start of the conference JNIC 2019. In *V Jornadas Nacionales de Investigación en Ciberseguridad*.

[173] Sanfilippo, S. (2006). hping. http://www.hping.org/. [Online; accessed 18-Oct-2019].

[174] Sanfilippo, S. (2019). hping3 Package Description. KALI Tools. https://tools.kali.org/information-gathering/hping3. [Online; accessed 18-Oct-2019].

[175] Schoenwaelder, J. (2008). Simple Network Management Protocol (SNMP) Context EngineID Discovery. Technical report, Jacobs University Bremen, https://tools.ietf.org/html/rfc5343. [Online; accessed 27-Nov-2019].

[176] Schulze, E. (2019). Twitter and Facebook could be facing billions in fines after Ireland investigations. https://www.cnbc.com/2019/10/07/facebook-twitter-investigations-in-ireland-reach-conclusion.html. [Online; accessed 31-Oct-2019].

[177] Security Onion Solutions (2008). Security Onion. https://securityonion.net/. [Online; accessed 20-Ago-2019].

[178] Seif, G. (2018). The 5 Clustering Algorithms Data Scientists Need to Know. https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

[179] Skansi, S. (2018). *Introduction to Deep Learning. From Logical Calculus to Artificial Intelligence*. Springer, 1st edition.

[180] Smilde, A. K. and Mechelen, I. V. (2019). *Data Fusion Methodology and Applications*, volume 31, chapter 2. A Framework for Low-Level Data Fusion, pages 27–50. Elsevier.

[181] Splunk (2005). Use splunk app for infrastructure. https://docs.splunk.com/Documentation/InfraApp/2.0.0/User/Alerts. [Online; accessed 25-Nov-2019].

[182] Stallings, W. (2014). *Data and computer communications*. Boston, https://us.norton.com/internetsecurity-privacy-data-breaches-what-you-need-to-know.html, 8 edition. [Online; accessed 18-Oct-2019].

[183] Symantec (2019). What is a data breach? Technical report, Symantec.

[184] Techopedia (2019a). Computer Emergency Response Team (CERT). Technical report, Techopedia, https://www.techopedia.com/definition/31003/computer-emergency-response-team-cert. [Online; accessed 18-Ago-2019].

[185] Techopedia (2019b). Computer Security Incident Response Team (CSIRT). Technical report, Techopedia, https://www.techopedia.com/definition/24837/computer-security-incident-response-team-csirt. [Online; accessed 18-Ago-2019].

[186] Tenable (1988). Nessus. https://www.tenable.com/lp/campaigns/19/try-nessus/?&utm_source=google&utm_medium=cpc&utm_term=nessus&utm_content=39801
hv-brand&gclid=EAIaIQobChMI65_RiZWZ5gIVQoXVCh0JEwcMEAAYASAAEgK9mv
[Online; accessed 20-Ago-2019].

[187] **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016a). Comparación de métodos de diagnóstico de anomalías en monitorización estadística multivariante de redes. In *Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Menorca (Spain)*.

[188] **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016b). Diagnóstico de Anomalías: Gráficos de Contribución vs oMEDA. In *I Jornadas de Investigadores en Formación Fomentando la interdisciplinariedad (JIFFI), Granada (Spain)*.

[189] **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2016c). Fault Diagnosis: Contribution plots vs oMEDA. In *XVI Chemometrics in Analytical Chemistry (CAC), Barcelona (Spain)*.

[190] **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2017a). Defending the network. Detection and Diagnosis of Anomalies. In *CITIC-Coffees, Granada (Spain)*.

[191] **Fuentes-García, N. M.**, Camacho, J., and Maciá-Fernández, G. (2019a). Evaluación de mejoras en la monitorización estadística multivariante para la detección de anomalías en tráfico ciclo-estacionario. In *V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), Cáceres (Spain)*.

[192] **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019b). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. *Journal of Chemometrics*, 33(11).

[193] **Fuentes-García, N. M.**, González-Martínez, J. M., Maciá-Fernández, G., and Camacho, J. (2019c). PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control. In *Scandinavian Symposium on Chemometrics (SSC16), Oslo (Norway)*.

[194] **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2017b). A Univariate Approach for Diagnosis in PCA-MSPC. In *Scandinavian Symposium on Chemometrics (SSC15), Naantali (Finland)*.

[195] **Fuentes-García, N. M.**, Maciá-Fernández, G., and Camacho, J. (2018). Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control. *Chemometrics and Intelligent Laboratory Systems*, 172:194–210.

[196] The Guardian (2019). Facebook to pay 5bn Dollars fine as regulator settles Cambridge Analytica complaint. https://www.theguardian.com/technology/2019/jul/24/facebook-to-pay-5bn-fine-as-regulator-files-cambridge-analytica-complaint. [Online; accessed 18-Oct-2019].

[197] Thuraisingham, B., Kantarcioglu, M., Bertino, E., and Clifton, C. (2017). Towards a Framework for Developing Cyber Privacy Metrics: A Vision Paper. In *IEEE 6th International Congress on Big Data*, pages 256–265.

[198] Tracy, N. D., Young, J. C., and Mason, R. L. (1992). Multivariate control charts for individual observations. *Journal of Quality Technology*, 24(2):88–95.

[199] TrendMicro (2018). Data Breaches 101: How They Happen, What Gets Stolen, and Where It All Goes. Technical report, Trend Micro, https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/data-breach-101. [Online; accessed 15-Ago-2019].

[200] Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley series in behavioral science: quantitative methods 7616. Addison Wesley.

[201] Undey, C., Ertun, S., and Cinar, A. (2003). Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis. *Industrial & Engineering Chemistry Research*, 42(20):4645–4658.

[202] UTC-students (2008). Gephi. https://gephi.org/. [Online; accessed 10-Sep-2019].

[203] van Sprang, E., Ramaker, H., Westerhuis, J., Gurden, S., and Smilde, A. (2002). Critical evaluation of approaches for on-line batch process monitoring. *Chemical Engineering Science*, 57(18):3979–3991.

[204] Vandoorselaere, Y. (2005). Prelude. https://www.prelude-siem.com/en/author/otran/page/2/. [Online; accessed 18-Ago-2019].

[205] Varonis (2019). 60 must-know cybersecurity statistics for 2019. online. Updated: 4/17/2019.

[206] Verizon (2019). 2019 Data Breach Investigations Report. Technical report, https://enterprise.verizon.com/resources/reports/dbir/. [Online; accessed 10-Ago-2019].

[207] Vincent, A., Barger, R., Pendergast, A., and Reichel, L. (2011). Threat Connect. https://threatconnect.com/. [Online; accessed 18-Oct-2019].

[208] Visscher, B. (2014). Sguil. https://sourceforge.net/projects/sguil/. [Online; accessed 18-Ago-2019].

[209] Vormayr, G., Zseby, T., and Fabini, J. (2016). Botnet Communication Patterns. *Communications Surveys and Tutorials*, 19(4):2768–2796.

[210] Wagner, I. and Eckhoff, D. (2018). Technical Privacy Metrics: A Systematic Survey. *ACM Computing Surveys*, 51(57).

[211] Wazuh Inc. (2019). The Open Source Security Platform. https://wazuh.com/. [Online; accessed 18-AgoOct-2019].

[212] Westerhuis, J. A., Gurden, S. P., and Smilde, A. K. (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51:95–114.

[213] Wise, B. M., Ricker, N. L., Veltkamp, D. F., and Kowalski, B. R. (1990). Theoretical basis for the use of principal component models for monitoring multivariate processes. *Process Control and Quality*, 1(1):41–51.

[214] Wold, S., Kettaneh, N., Friden, H., and Holmberg, A. (1998). Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, 44(1-2):331–340.

[215] Wold, S., Kettaneh-Wold, N., MacGregor, J. F., and Dunn, K. G. (2009). Batch Process Modeling and MSPC. *Elsevier: Oxford*, 1:163–195.

[216] Xia, H., Fang, B., Roughan, M., Cho, K., and Tune, P. (2018). A Basis Evolution framework for network traffic anomaly detection. *Computer Networks*, 135:15–31.

[217] Yiu, T. (2012). Understanding Random Forest. How the Algorithm Works and Why it Is So Effective. Technical report, Towards Data Science. Sharing concpets, ideas and codes., nderstanding Random Forest. How the Algorithm Works and Why it Is So Effective. [Online; accessed 18-Oct-2019].

[218] Yu, Z. and Tsai, J. J. P. (2011). *Intrusion Detection. A Machine Learning Approach*. Imperial College Press (Series in Electrical and Computer Engineering Vol. 3).

"No TV and no beer make Homer go crazy."

Homer, The Simpsons (originally from Jack Torrance, The Shining, 1980)

A

List of Acronyms

ANOVA ANalysis Of VAriance

AOC Abnormal Operation Condition

AS Auto-Scaling

AUC Area Under the Curve

BMSPC Batch MSPC

CL Control Limit

CP Contribution Plots

CERT Computer Emergency Response Team

CISO Chief Information Security Officer

CSIRT Computer Security Incident Response Team

CVE Common Vulnerabilities and Exposures

DM Data Mining

DD Data-Driven

DDoS Distributed Denial of Service

DoS Denial of Service

EDA Exploratory Data Analysis

EWMA Exponentially Weighted Moving Average

EWMW Exponentially Weighted Moving Window

FIM File Integrity Monitoring

GUI Graphical User Interface

HIDS Host IDS

IDMEF Intrusion Detection Message Exchange Format

IDS Intrusion Detection System

INCIBE Instituto Nacional de Ciberseguridad

IP Internet Protocol

IPS Intrusion Prevention System

ISP Internet Server Provider

IT Information Technology

KNN K-nearest neighbors

LSD Least Significant Difference

MBDA Multivariate Big Data Analysis

MEDA Multivariate Exploratory Data Analysis

ML Machine Learning

MSNM Multivariate Statistical Network Monitoring

MSPC Multivariate Statistical Process Control

MVBatch MultiVariate Batch

NIDS Network IDS

NIST National Institute of Standards and Technology

NOC Normal Operation Condition

NSM Network Security Monitoring

NSD Normalized Squared Difference

NTP Network Time Protocol

NVD National Vulnerability Database

oMEDA observation-based Missing-data method for Exploratory Data Analysis

OSSEC Open Source HIDS SECurity

OSI Open Systems Interconnection

OSSIM Open Source Security Information Management

OTI Overall Type I

OTII Overall Type II

PARAMO PARAmeters from More Observations

PC Principal Component

PCA Principal Component Analysis

PLS Partial Least Squares

RADAF RAw DAta Filtering

RBC Reconstruction-Based Contributions

RGTW Relaxed Greedy Time Warping

ROP Ratio number-of-Observations-to-the-number-of-Parameters

ROC Receiver Operating Characteristics

SEM Security Event Management

SIEM Security Information and Event Management

SIM Security Information Management

SME Small and Medium-sized Enterprise

SNMP Simple Network Management Packet

SPC Statistical Process Control

SSR Sum of Squared Residuals

SVM Support Vector Machines

SVD Singular Value Decomposition

TCP Transmission Control Protocol

TCS Trajectory Centering and Scaling

UBA User Behavior Analytics

UCL Upper Control Limit

UDP User Datagram Protocol

U-PARAMO Uniform PARAMO

U-RADAF Uniform RADAF

U-Squared Univariate Squared

UTM Universal Threat Management

UWMW Uniformly Weighted Moving Window

VCS Variable Centering and Scaling

VR Virtual Router

X-PARAMO eXponential PARAMO

X-RADAF eXponential RADAF

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

Marie Curie, Nobel Prize in Physics in 1903

B

Related Terms

This appendix presents a compilation of some useful definitions that can be found in the related literature. These terms will help to an easier reading by having the same concepts in mind.

Threat. A **threat** is any situation or event that may damage a system or network. This refers to unveiling, destroying, modifying or denying the access to the data or services [16]. A threat that comes true is termed an **attack**.

Asset. An **asset** is any valuable element in a given framework. In IT Security, it can refer to machines or physical resources, to certain information or intellectual property, and also prestige or reputation. The value of an asset can

be estimated as the time and resources needed (cost) for replacement or for returning the asset to its original state [19, 82].

Security Operator. A **security operator** is a person in charge of administering and monitoring the security system in an organization, while a **security analyst** is in charge of analyzing and discovering *vulnerabilities* and *risks* in the organization. Both security operators and analysts are usually interchangeable terms, and they are people who take part of the **security team**.

Chief Information Security Officer (CISO). A CISO is a person who has the highest responsibility in relation to IT Security of an organization. Some of the tasks corresponding to the CISO are supervise the planning and deployment of everything related to the information security, including the definition of security policies and the design of infrastructures for supporting the prevention, detection and response of any IT Security issue.

Security Team. Security Team are people in charge of the security of an organization, including both security analysts and operators. See also CERT.

Computer Emergency Response Team (CERT). CERTs are groups of security specialists that aim to detect and respond to cybersecurity incidents, warning and/or advising the rest of the citizens and organizations about them. CERTs are typically dependant either on governmental organizations or private big companies [72, 184]. The CSIRTs are frequently considered to be equivalent to the CERTs. However, they are usually more focused in detection and response, rather than in prevention [169, 185]. A list with the European CERT and CSIRT entities can be found in [73].

Vulnerability. A **vulnerability** is an *asset* that could lead to its unauthorized exploitation. A vulnerability may exist due to a bad design, implementation or even for intentional reasons [16, 19].

Asset. An **asset** is any valuable element in a given framework. In IT Security, it can refer to machines or physical resources, to certain information or intellectual property, and also prestige or reputation. The value of an asset can be estimated as the time and resources needed (cost) for replacement or for returning the asset to its original state [19, 82].

Risk. A **risk** is the probability of suffering any damage or lost. Its value can be considered a combination of the threat, vulnerability and relevance of the asset [19, 82].

IT Security Event. An **IT Security event** in a system or network refers to any undesired situation or modification in the system or network that occurs for a period of time and that is susceptible to be detected by the security system. If the event is detected, this usually generates an **alert**, which is recorded as an individual log as a part of a file or database.

Prioritization. This is also known as *triage* and it allows to determine the order in which the events are analyzed and/or solved. Prioritization is based on one or more criteria. These criteria may be, for example, the impact or the magnitude of the alert.

(IT Security) Incident. An **IT security incident** happens when the IT Security of an organization or company, of an organization or company is compromised due to any reason, violating any of the IT Security requirements [16, 218]. One of the best known IT Security incidents are the intrusions, which aim is just to compromise the confidentiality, integrity or

availability of a resource, as well as avoiding the security mechanisms in a network or system [16, 20, 218].

Data Breach. A **data breach** is a type of security incident that occurs when someone accesses and extracts personal or confidential information without any authorization [183, 199].

Malware. **Malware** refers to any type of software that is built with the aim of damaging any device (*e.g.* computer or telephone). It is used with different purposes, such as stealing information or denying access to legitimate users [15, 133].

Correlation. In the context of IT Security, the term **correlation** means finding connections among distinct data sources or IT Security events, rather than being used with the traditional statistical meaning.

Zero-Day* Attack.** A ***zero-day attack is an attack that had not been previously seen and, thus, its features and signature are not known.

Sensor. A **sensor** is a mechanism that collects data from the network, generating logs or records that can be analyzed by the security team. Sensors are composed of *collectors* and, sometimes, *processors*, which allow to capture and transform the information, respectively, prior to send it to the integrator module. However, the most simple sensors might only be composed of a collector module.

Integrators. **Integrators** combine the data collected by the sensors and detect intrusions in them. First, the different records are *correlated* to extend their semantic information, yielding good models for detection of attacks or abnormal activities. This requires pre-processing the format of the data to be

readable by the **correlation engine**. Afterwards, the **detection engine** detects illegitimate network traffic by means of either a model of normal operation or the signatures for known attacks.

Sniffer. A **sniffer** is a program that collects and analyzes packets in a communication network.

Intrusion Detection System (IDS). Intrusion Detection Systems (IDSs) are a set of techniques to detect suspicious activity (possible intrusions) by monitoring and analyzing the events in a network or a device [62, 103, 218]. These are a particular type of sensor, which are composed of collector, processor and detection engine. When these systems also allow to deploy defensive responses to the attacks, they are called **IPSs**.

Security Event Management (SEM). SEMs systems are in charge of "*the collection, analysis and escalation of indications and warnings to detect and respond to intrusions*" [20]. Its aim is visualize and understand traffic data by using a single and unified tool that combines different data sources.

Security Information Management (SIM). SIM systems allow the regulatory compliance, analysis and notification of the events, as well as long-time storage of such events. This makes it possible to perform forensic analysis once an attack has taken place.

Security Information and Event Management (SIEM) . SIEM systems are the combination of the SIM and SEM systems. Thus, the objective of a SIEM system is to aggregate and analyze the information collected from a number of *sensors* to detect, select, classify and validate incidents in a network [112]. In addition, a SIEM system generates reports related to the compliance of security policies, useful to pass audits. SIEM systems allow

the visualization and prioritization of the events, thus helping the security operators to interpret and understand the alarms [87, 103].

Universal Threat Management (UTM). This is a type of "*multi-function network security product used by small or midsize business*" [86]. These devices have high level functionalities (multi-function gateway), which can be, for example, a firewall in the application layer of the TCP/IP and OSI models, Intrusion Prevention and Detection (IPS and IDS), antivirus, anti-spam and anti-phishing [32, 82, 182]. The main advantages of the UTMs are their reduced cost and complexity, while the drawbacks are that UTMs usually cannot correlate events.

Parsing. **Parsing** is the process of identifying and extracting individual parts that compose a log to obtain a logical and organized data structure [135].

Pivoting. **Pivoting** refers to the ability of going from one data source to another.

Feature Extraction. **Feature extraction** consists on obtaining new variables by transforming the original data records.

Observation. An **observation** is the set of properties or features that are measured for an entity. The entities of interest can be disparate (*e.g.* time intervals or devices).

Key Process Event. A **key process event** defines the moment in which each step of a process takes place (when the step starts and ends). Key process events usually vary from batch to batch (see Fig. 3.3 (a)).

"Shut up brain. Now I have friends, I don't need you anymore."

Lisa, The Simpsons

C

Oversights on the Application of RBC for the D-statistic

The RBC expression for the *Hotelling's* T^2 is analysed here. $T^2 = \mathbf{x}' \cdot \mathbf{D}_A \cdot \mathbf{x}$, with $\mathbf{D}_A = \mathbf{P}_A \cdot \Lambda_A^{-1} \cdot \mathbf{P}'_A$, is used to define the RBC expression for the D-statistic in Alcala and Joe Qin [4], Alcala and Qin [5]. Following a similar derivation procedure as in Camacho [28], we can define:

$$\breve{\alpha}_m^A = \sum_{a=1}^{A} \frac{p_{m,a}^2}{\Lambda_a} \tag{C.1}$$

$$\check{\beta}^{A}_{v,m} = \sum_{a=1}^{A} \frac{p_{v,a} \cdot p_{m,a}}{\Lambda_a} \tag{C.2}$$

where $p_{m,a}$ is the loading of the variable m and the selected component a, Λ_a is the element corresponding to the selected component a on the main diagonal of Λ_A, $p_{v,m}$ is the loading of variable v, and $\check{\beta}_{v,m}$ are the elements that do not belong to the main diagonal of the matrix.

Let us consider now Equation (C.3) to be the expression in the D-statistic for the variable m

$$\mathbf{i}_m \cdot \mathbf{D}_A \cdot \mathbf{x} = \check{\alpha}^{A}_{m} \cdot x_m + \sum_{v \neq m} \check{\beta}^{A}_{v,m} \cdot x_v \tag{C.3}$$

and

$$d_{m,m} = \mathbf{i}'_m \cdot \mathbf{D}_A \cdot \mathbf{i}_m = \check{\alpha}^{A}_{m} \tag{C.4}$$

the element $d_{m,m}$ corresponding to the diagonal of the matrix $\mathbf{D_A}$. From equations (7.3) and (C.4):

$$rbc^{D}_{m} = \mathbf{x}' \cdot \mathbf{D}_A \cdot \mathbf{i}_m \cdot (\mathbf{i}'_m \cdot \mathbf{D}_A \cdot \mathbf{i}_m)^{-1} \cdot \mathbf{i}'_m \cdot \mathbf{D}_A \cdot \mathbf{x} \tag{C.5}$$

is the extended form of Equation (7.3) for RBC. By combining it with Equation (C.3), it can be re-written as follows:

$$\begin{aligned} rbc^{D}_{m} = {} & \frac{(\check{\alpha}^{A}_{m})^2 \cdot x_m^2 + \sum_{v \neq m} (\check{\beta}^{A}_{v,m})^2 \cdot x_v^2}{\check{\alpha}^{A}_{m}} + \\ & \frac{2 \cdot \check{\alpha}^{A}_{m} \cdot x_m \cdot \sum_{v \neq m} \check{\beta}^{A}_{v,m} \cdot x_v}{\check{\alpha}^{A}_{m}} + \\ & \frac{2 \cdot \sum_{v \neq m} \sum_{w \neq v \neq m} \check{\beta}^{A}_{v,m} \cdot \check{\beta}^{A}_{w,m}}{\check{\alpha}^{A}_{m}} \end{aligned} \tag{C.6}$$

By applying Equation (C.6) for 1 selected PC, and replacing $\breve{\alpha}_m^A$ and $\breve{\beta}_{v,m}^A$ with Equations (C.1) and (C.2), the Equation (C.7) is obtained:

$$rbc_m^D = ((\frac{p_{m,1}^2}{\Lambda_1})^2 \cdot x_m^2 + \sum_{v \neq m} (\frac{p_{m,1} \cdot p_{v,1}}{\Lambda_1})^2 \cdot x_v^2 + 2 \cdot \frac{p_{m,1}^2}{\Lambda_1} \cdot x_m \cdot \sum_{v \neq m} \frac{p_{m,1} \cdot p_{v,1}}{\Lambda_1} \cdot x_v + 2 \cdot \sum_{v \neq m} \sum_{w \neq v \neq m} \frac{p_{m,1}^2 \cdot p_{v,1} \cdot p_{w,1}}{\Lambda_1} \cdot x_v \cdot x'_w) \cdot \frac{1}{p_{m,1}^2/\Lambda_1} \quad \text{(C.7)}$$

By grouping and simplifying Equation (C.7) in Equations (C.8) and (C.9),

$$rbc_m^{D_{1PC}} = \frac{1}{\Lambda_1} \cdot p_{m,1}^2 \cdot x_m^2 + \frac{1}{\Lambda_1} \cdot \sum_{v \neq m} p_{v,1}^2 \cdot x_v^2 + \frac{2}{\Lambda_1} \cdot x_m \cdot \sum_{v \neq m} p_{m,m} \cdot p_{v,1} \cdot x_v + \frac{2}{\Lambda_1} \cdot \sum_{v \neq m} \sum_{w \neq v \neq m} p_{v,1} \cdot p_{w,1} \cdot x_v \cdot x'_w \quad \text{(C.8)}$$

$$rbc_m^{D_{1PC}} = \frac{1}{\Lambda_1} \cdot \sum_v p_{v,1}^2 \cdot x_v^2 + \frac{2}{\Lambda_1} \cdot \sum_v \sum_{w \neq v} p_{v,1} \cdot p_{w,1} \cdot x_v \cdot x'_w = rbc_v^{D_{1PC}} \quad \text{(C.9)}$$

it is shown that the RBC value for the expression in the D-statistic is exactly the same for every variable, *i.e.*, each variable has the same contribution, which makes, according to Equation (7.18), the ratio $\gamma = 1$. This is translated

into a lack of diagnosis ability for RBC for the D-statistic if 1 PC is selected, as it cannot be distinguished which variables are affected when there is an anomaly.

"Shut up brain. Now I have friends, I don't need you anymore."

Lisa, The Simpsons

D

Efficiency Calculation for Standard and Hierarchical Fusion

We evaluate theoretically the computational cost of the hierarchical approach in relation to the standard fusion of the data [129]. Recall that for a matrix $\mathbf{X}$ of N observations by M variables, the PCA model is calculated following Eq. (3.3). The complexity of this operation [88] is obtained from the complexity of calculating the covariance matrix, $\mathcal{O}(N \cdot M^2)$, and its eigenvalue decomposition, $\mathcal{O}(M^3)$, following:

$$\mathscr{C}_S = \mathcal{O}(NM^2 + M^3) \approx \mathcal{O}(NM^2), \quad \text{for} \quad N \gg 0 \tag{D.1}$$

When the calculation of the PCA model is split as explained in Chapters 4 and 8 for the hierarchical approach, we are actually computing L local PCA models, every one with M/L variables, plus one global model with $L \cdot 2$ variables. For this reason, the complexity now becomes:

$$\mathscr{C}_H \approx \mathscr{O}(N\left[\frac{M^2}{L} + L^2\right]), \quad \text{for } N \gg 0 \tag{D.2}$$

Thus, the computation load is lower in the hierarchical than in the standard fusion if the number of nodes involved in the computation, L, and the number of variables considered, M, comply that $M > L(1-1/L)^{-\frac{1}{2}} \approx L$ (for $L \gg 0$).

We can conclude that the computation needed for the algorithm is even lower than in standard PCA when the number of sensors L is lower than the number of variables M considered in the computation. This actually corresponds to regular scenarios, as the number of variables is usually high in networking environments.

"¡Todo está saliendo a pedir de Milhouse!"

Milhouse, Los Simpsons

E

Resumen amplio en castellano

Resumen amplio en castellano

Motivación

Un **incidente de seguridad** (o incidente IT, del inglés *Information Technology*) sucede cuando la seguridad de una organización o compañía se ve comprometida debido a cualquier razón, violando cualquiera de los requerimientos típicos de seguridad IT (*Confidencialidad*, *Integridad*, *Disponibilidad*, *Responsabilidad*, *No repudio*, *Autenticación* y *Autorización* [16, 82, 218]). Un problema importante cuando se afrontan las amenazas de seguridad de la información es que el tiempo requerido para comprometer un sistema o red es realmente reducido (del orden de segundos o minutos) si se compara con el tiempo necesario para la detección y reacción frente a un ataque por parte del personal de seguridad IT (que puede llevar desde días hasta meses) [84]. Por eso, es muy importante reducir el tiempo de detección y respuesta. Además, el personal de seguridad IT con frecuencia recibe más alarmas de las que puede manejar en su jornada laboral [9, 62, 135]. En este sentido, es deseable que los mecanismos de detección también permitan la adecuada *priorización* de las alarmas. La **priorización** de eventos se basa en uno o más criterios, como pueden ser la magnitud o el impacto del evento.

La seguridad de la red es una parte esencial de la ciberseguridad y la seguridad de la información. Su objetivo es hacer que las infraestructuras de comunicaciones cumplan con todos los requisitos de seguridad mencionados anteriormente. Hay distintos puntos de vista para tratar la seguridad en la

red. Según el objetivo principal que se afronta, estos enfoques se pueden clasificar en: *prevención*, *detección* y *respuesta*. Dichos enfoques no son excluyentes, al contrario, se suelen aplicar de forma conjunta para lograr una mayor seguridad en la red [16, 19, 82]. La Fig. 1.2 (Capítulo 1) muestra la clasificación de estos enfoques, así como algunos ejemplos de soluciones que se pueden clasificar en esos grupos. Así, cuando hablamos sobre **prevención**, algunas de las herramientas más conocidas son los **cortafuegos**, los **antivirus** y el **cifrado de datos**. Por su parte, algunas de las herramientas de **detección** (y frecuentemente también de **respuesta**) más conocidas son los **IDS** (del inglés, *Intrusion Detection System*), que permiten detectar actividades sospechosas mediante la monitorización y análisis de los eventos de la red o dispositivos [62, 103, 218]; y los **SIEM** (del inglés, *Security Information and Event Management*), que permiten agregar y analizar la información recopilada de varios *sensores* para detectar, seleccionar, clasificar y validar los incidentes de una red, así como realizar informes para superar auditorías de seguridad [112].

NSM (del inglés, *Network Security Monitoring*) es un enfoque que pretende detectar los ataques en una red mediante la monitorización del tráfico de dicha red [134, 218]. Esto se lleva a cabo mediante la captura, correlación y análisis de dicho tráfico, para la detección de intrusiones [20]. En ocasiones, NSM también implica proporcionar respuestas o tomar acciones cuando se detecta el ataque. El objetivo de NSM es dar visibilidad a los eventos de la red. Se puede considerar que los sistemas SIEM siguen este enfoque [19, 20, 32].

En general, el método de detección se puede agrupar en *basado en firmas* y *basado en detección de anomalías* [68, 81, 103]. El primero identifica ataques a partir de patrones previamente definidos. El segundo detecta desviaciones del comportamiento normal en una red o sistema en relación a un modelo previamente entrenado. Los sistemas basados en firmas no pueden detectar nuevos ataques (también llamados ataques *zero-day*) mientras que los basados en detección de anomalías sí podrían detectarlos. Un ataque ***zero-day*** es un

ataque que no había sido visto antes y, por tanto, sus características y firma son desconocidas. Los enfoques basados en detección de anomalías tienden a generar un elevado número de (falsas) alarmas, lo que puede llegar a ser un problema. Así, uno de los principales retos para la detección es lograr un equilibrio entre ambas propiedades [62, 218].

Esta tesis doctoral se centra en los sistemas basados en la detección de anomalías en el contexto de NSM. Más concretamente, el propósito de este trabajo es impulsar los recientes desarrollos en análisis multivariante de datos en NSM [42] y proponer alternativas para mejorar estas técnicas.

Monitorización estadística multivariante

MSPC (del inglés, *Multivariate Statistical Process Control*) se desarrolló originalmente para reducir la variabilidad en los productos y/o procesos industriales. El propósito de esta metodología es distinguir entre causas asignables y causas comunes de variación en un proceso. Esencialmente, esto significa discriminar entre eventos cuya causa es identificable y resoluble y aquellos que se deben a un suceso normal en el proceso. Utilizar MSPC permite la monitorización simultánea de varias variables mediante la consideración de sus correlaciones para obtener un mejor modelo y, así, una mejor detección de anomalías. Debido a la alta dimensionalidad de las variables en estos procesos, es frecuente aplicar técnicas basadas en variables latentes para reducir las dimensiones, como **PCA** (del inglés, *Principal Component Analysis*).

PCA-MSPC permite la monitorización de un par de estadísticos complementarios que posibilitan la monitorización indirecta de un alto número de variables. Los estadísticos se calculan a partir de la descomposición PCA de los datos de calibración previamente pre-procesados (típicamente mediante centrado en la media o auto-escalado) para consturir un modelo de operación normal (fase I) [117, 150, 213]. Esta metodología se aplica para detectar si el comportamiento de los nuevos datos encaja con el modelo previamente

ajustado (fase II). Si no es así, indica que los datos son anómalos y que es necesario llevar a cabo un paso adicional, el **diagnóstico**, que permite identificar las variables que están relacionadas con la anomalía y ayuda a los analistas a encontrar la causa real de dicha anomalía. Se pueden encontrar más detalles sobre MSPC y PCA en el Capítulo 3 y en [42, 76, 150].

Lakhina *et al.* propusieron por primera vez el uso de PCA para la detección de anomalías en tráfico de red [118]. Desde entonces, se han propuesto varias modificaciones para resolver algunos de los problemas identificados en dicho enfoque [26, 27, 66, 168]. En los últimos años, otros trabajos de investigación en análisis multivariante relacionados con seguridad de la información también han combinado PCA con otras técnicas [3, 54, 74, 109, 216]. Sin embargo, la mayoría de estos enfoques todavía mantienen parte de los problemas descubiertos en relación al trabajo original de Lakhina *et al.* y que son debidos, principalmente, a sus diferencias con respecto a la teoría de MSPC [42]. Esto motivó el desarrollo de la metodología **MSNM** (del inglés, *Multivariate Statistical Network Monitoring*), que es una extensión de MSPC para la monitorización del tráfico de red, y que se introdujo en 2015 [42].

MSNM permite combinar datos de tráfico con otras fuentes de datos de seguridad [30], y ha demostrado tener una capacidad de detección comparable a las metodologías de aprendizaje automático del estado del arte [38], con la ventaja de que, además, permite llevar a cabo un diagnóstico una vez detectada la anomalía [36, 38].

Al igual que otras metodologías de aprendizaje automático, y a diferencia de MSPC, MSNM necesita realizar pasos de *parsing* y fusión. Esto se debe a que los datos de red proceden de registros de distintos sensores y en distintos formatos, por lo que es necesario procesar y transformar los datos para que tengan un formato uniforme e interpretable. Así, MSNM consta de cuatro pasos: *1) Parsing*, *2) Fusión*, *3) Detección*, *4) Diagnóstico*. Durante los últimos años desde su primera propuesta, metodología MSNM ha sido extendida, proponiendo mejoras en los pasos existentes, así como añadiendo

otros nuevos [33, 36, 40, 50, 129, 192, 195]. Todas ellas se encuentran organizadas en el esquema que se muestra en la Fig. 4.6 (Capítulo 4). Esta tesis ha contribuido a la mayoría de estas extensiones, liderando aquellas relacionadas con el pre-procesamiento (enfoque PARAMO - del inglés, *PARAmeters from More Observations*) [192] y el diagnóstico (metodología de comparación de métodos de diagnóstico y método U-Squared - del inglés, *Univariate Squared*) [195], que se detallan como contribuciones principales de esta tesis en los Capítulos 6 and 7, respectivamente. A continuación, se describen dichas contribuciones de forma resumida.

Contribuciones

Los resultados de esta investigación han sido compartidos con la comunidad científica mediante la participación en distintas conferencias tanto de ámbito internacional [94, 189, 193, 194] como nacional [131, 132, 187, 191]. Extos resultados también han sid opublicados en revistas de alto impacto científico [36, 40, 192, 195].

Pre-procesamiento: PARAMO

El pre-procesamiento es esencial para construir un buen modelo de calibración en MSNM. Es importante centrar los datos para detectar desviaciones respecto a la media. Además, escalar los datos de red también es necesario porque, en general, estos datos son heterogéneos, ya que proceden de distitas fuentes y presentan distintos formatos, así como las variables suelen presentar dispersiones muy distintas entre sí.

Los **procesos por lotes** son un tipo especial de procesos industriales muy importantes para distintas áreas [148–150]. Un **lote** se refiere a un proceso compuesto de distintas fases y pasos que se repiten de manera cíclica, de forma similar a como lo hace una receta de cocina [52]. El tráfico de red también se

caracteriza por la presencia de ciclos en su actividad. Esta particularidad se denomina ciclo-estacionariedad [52, 115, 148–150]. Por ejemplo, se pueden observar patrones similares durante el día o la noche, pero también entre días laborables o fines de semana. Estos patrones se repiten de forma periódica para una misma red. Para considerar la ciclo-estacionariedad de los datos en el modelado matemático, estos se organizan en tres dimensiones: $I \times J \times K$. Los principales métodos de pre-procesamiento en BMSPC (del inglés *Batch MSPC*) son TCS (del inglés, *Trajectory Centering and Scaling*) [149] y VCS (del inglés, *Variable Centering and Scaling*) [214]. Un problema que presenta TCS es la incertidumbre en la estimación de las medias y, especialmente, la de las desviaciones estándar. Esto se debe a que el número de muestras utilizado para estimar los parámetros (medias y desviaciones) es reducido [92]. La principal ventaja de este tipo de pre-procesamiento es la posibilidad de considerar la naturaleza cíclica de los datos de red, permitiendo construir mejores modelos.

Dado que TCS es un enfoque más sensible para detectar variaciones de la normalidad en el tiempo (considera la ciclo-estacionariedad) y, por tanto, más confiable como método de detección, en el Capítulo 6, se proponen distintas alternativas de pre-procesamiento basadas en TCS: **PARAMO** (del inglés, *PARAmeters from More Observations*) y **RADAF** (del inglés, *RAw DAta Filtering*). El objetivo es reducir la incertidumbre introducida durante la estimación de parámetros incrementando el número de observaciones utilizadas para dicha estimación, a la par que se mantiene la capacidad de modelar la ciclo-estacionariedad de los datos. PARAMO calcula los parámetros de pre-procesamiento en un instante de tiempo k considerando una serie de observaciones en torno al instante de tiempo dado, mientras que RADAF filtra los datos originales para "suavizarlos" antes de aplicar TCS. Ambos enfoques se basan en un esquema de ventana deslizante en el que los datos son ponderados de una entre dos formas distintas: exponencial o uniforme. Teniendo en

cuenta los problemas expuestos en las Secciones 6.5.1 y 6.5.2, se recomienda la aplicación de PARAMO utilizando ventanas simétricas [192].

Para validar la propuesta, se realizó un estudio exhaustivo, comparando primero PARAMO con RADAF y, posteriormente, PARAMO con el método de pre-procesamiento tradicional, TCS. Para ello se llevaron a cabo varios ANOVA (del inglés, *ANalysis Of VAriance*). El proceso de cultivo del *Saccaromyces Cerevisiae* se utilizó generar varios conjuntos de datos, con el objetivo de validar la mejora de la estabilidad paramétrica así como la capacidad de detección de fallos. El estudio comparativo reveló que PARAMO mejora respecto a la metodología establecida para el pre-procesamiento de datos por lotes. El trabajo completo está publicado en [192].

Diagnosis: Comparación de métodos de diagnóstico y U-Squared

Una de las acciones más importantes en los sistemas de monitorización MSPC y MSNM es, una vez detectada una anomalía, identificar las variables relacionadas con dicha anomalía. A esto se le denomina diagnóstico [117] (ver Capítulo 7 para una revisión de la literatura relacionada). Algunos métodos multivariantes de diagnóstico son CP (del inglés, *Contribution Plots*), RBC (del inglés, *Reconstruction Based Contributions*), y oMEDA (del inglés, *observation-based Missing-data method for Exploratory Data Analysis*). Desde una perspectiva general, los trabajos revisados introducen métodos nuevos y proporcionan comparaciones limitadas con varios métodos de referencia. Por otra parte, los métodos de diagnóstico multivariante suelen presentar un problema, el denominado efecto de *smearing*: la dispersión de la contribución de las variables contaminadas por una anomalía a variables no afectadas. Este problema suele llevar a un diagnóstico erróneo [114, 212], lo que complica más aún el proceso de evaluación de la anomalía.

Una vez se produce una anomalía, la estructura de correlación en el modelo podría dejar de ser válida para esa anomalía y, por tanto, la división

en modelo/residuo encontrada para los datos de calibración ya no sería óptima para el diagnóstico. Si esto ocurre, podemos pensar que no tiene sentido calcular la contribución de las variables a cada estadístico de forma separada, pudiendo ser más interesante considerar todo el espacio de variables para llevar a cabo el diagnóstico. Dado que oMEDA [29] se calcula de igual forma tanto en el subespacio del modelo como en el del residuo, se puede extender para considerar el espacio completo de variables. Así, utilizando las Ecuaciones (7.8) y (7.9), se obtiene la expresión (7.10), que denominamos **U-Squared** (del inglés, *Univariate Squared*). Nótese que esta expresión se corresponde con un enfoque univariante, ya que solamente considera el valor original de cada variable, y no sus contribuciones. En la literatura se pueden encontrar otros enfoques univariantes [113, 114], pero no se ha probado su eficacia en una comparativa exhaustiva.

Para comprobar la validez del método propuesto, se llevó a cabo una comparativa entre los distintos métodos de diagnóstico. Para que esta comparativa fuera equitativa, se propuso un procedimiento que consta de tres pasos: *1) Generación de anomalías con diagnóstico conocido*, *2) Definición de una métrica para evaluar el rendimiento del diagnóstico*, *3) Diseño experimental*. Esta metodología es otra de las contribuciones de esta tesis. Dicha metodología se aplicó simulando varios conjuntos de datos aleatorios utilizando *simuleMV* [31]. Estos conjuntos abarcan un amplio rango de situaciones, en las que varían el número de variables, el número de observaciones o la correlación, entre otros factores. El estudio se realizó siguiendo el enfoque de Monte Carlo, permitiendo cubrir la generación de anomalías tanto univariantes como multivariantes. Además, los resultados se validaron utilizando dos conjuntos de datos tomados de entornos realistas: uno obtenido a partir del tráfico de una red de comunicaciones [129], y otro obtenido mediante la simulación del proceso de cultivo de *Saccharomyces Cerevisiae* [120].

El ANOVA llevado a cabo sobre los resultados de la comparación entre los métodos siguiendo la metodología anterior indicó que hay varios parámetros

relevantes para el diagnóstico: las dimensiones (filas x columnas) de la matriz de calibración el MSPC/MSNM, el método de diagnóstico, el estadístico (*D* or *Q*), el número de PC seleccionados para el modelo PCA, y el número de variables afectadas por la anomalía. Los resultados de ANOVA mostraron que U-Squared presenta diferencias estadísticamente significativas respecto al resto de métodos, mejorando el rendimiento de dichos métodos en el D-estadístico. Por esta razón, se propone adoptar un enfoque de diagnóstico univariante, aunque la detección se lleve a cabo de forma multivariante. Este trabajo se publicó en [195].

Evaluación de extensiones MSNM con datos reales

La última contribución de esta tesis es la aplicación de algunas extensiones de MSNM sobre datos de tráfico obtenidos de una red real. El conjunto de datos UGR'16 [130] se ha seleccionado con tal propósito. Este conjunto contiene una amplia captura de datos de un ISP que fueron recopilados durante cuatro meses en 2016 (ver Secciones 5.2.1 y 8.2.2). UGR'16 también incluye ataques introducidos de forma intencionada [130], permitiendo la evaluación de la capacidad de detección y diagnóstico de los enfoques MSNM. Más concretamente, durante el cuarto mes, se implementaron ataques de DoS, escaneo de puertos y *neris botnet*, etiquetados como **DoS**, **scan11**, **scan44** y **botnet**. La captura original contiene anomalías reales (tanto etiquetadas como descubiertas *a posteriori*. Dichas anomalías no han sido consideradas en nuestros experimentos. Además, para las pruebas se han considerado los diez primeros días de ataques, ya que hay dos ataques de *botnet* etiquetados que realmente no se insertaron en la captura de los datos. Lo habitual es que los datos procedan de distintos sensores, sin embargo en el conjunto UGR'16 los datos proceden de un único sensor, por lo que es necesario tener routers "virtuales" con sensores, *VR** (ver Secciones 4.2.2 y 8.2.2) para parte de nuestra experimentación. La Tabla 8.1 muestra el orden seguido para llevar

a cabo dicha asignación. Por otra parte, los ataques se distribuyen como se muestra en la Tabla 8.2.

Una vez los sensores han recopilado los datos y las características han sido extraídas, los datos se pueden fusionar de distintas formas (ver Sección 4.1.2). Por este motivo, la evaluación se divide en dos partes: por una parte, los experimentos se llevan a cabo para la fusión estándar de los datos; y, por otra parte, la aplicación de MSNM para la fusión jerárquica de los datos (**H**). Aunque la fusión jerárquica no es una contribución de esta tesis *per se*, se aplica por primera vez a datos reales de red en este trabajo. La fusión de datos estándar en una sola matriz se puede hacer de varias formas. En este trabajo se sugieren dos alternativas: *i) Concatenar* el valor de las características (**C**) y *ii) Agregar* el valor de las características (**A**). Este último solamente es posible si todas las fuentes son de la misma naturaleza (por ejemplo, dos routers del mismo fabricante). Ambas están representadas en la Fig. 8.1. Además, durante el paso de detección, se aplica como mejora el pre-procesamiento ciclo-estacionario (PARAMO y TCS). Los resultados obtenidos se comparan con los que proporciona el auto-escalado. Por último, para el diagnóstico se aplican oMEDA y U-Squared. Estas extensiones se aplican siguiendo el enfoque en 5 pasos recientemente propuesto también como extensión MSNM [36].

El enfoque de fusión *jerárquico* fue propuesto como extensión MSNM por primera vez en [129]. Para evaluarlo, se han seleccionado las alternativas que podrían llevar a resultados significativamente distintos, considerando aquellos previamente obtenidos para la fusión estándar (ver Sección 8.2.3). Así, la fusión jerárquica se evalúa aplicando fusión de tipo **C**, auto-escalado para el pre-procesamiento, y U-Squared para el diagnóstico.

La relación entre el número de falsos y verdaderos positivos es muy importante en la monitorización de redes, ya que el número de alarmas puede ser excesivo en este contexto. Una forma de medir esta relación es el AUC (del inglés, *Area Under the Curve*), que representa el área bajo la curva ROC (del inglés, *Receiver Operating Characteristics*) [97, 138]. Esta es una medida

típica de los clasificadores de una clase, donde se encuentran los IDS basados en detección de anomalías, y también para los clasificadores de dos clases. Un clasificador ideal tiene $AUC = 1$, mientras uno con capacidad de detección nula tiene $AUC \approx 0.5$ [8]. Nosotros utilizamos tanto AUC como curvas ROC para evaluar el rendimiento de las técnicas bajo estudio en el contexto del conjunto de datos UGR'16.

En los resultados obtenidos, las extensiones MSNM para el pre-procesamiento tienen un rendimiento comparable con AS. También se ha comprobado que U-Squared mejora el diagnóstico, contribuyendo a resolver el problema de *smearing* y reduciendo la complejidad de la diagnosis. Esto lleva también a los operadores de seguridad a centrarse en las características más relevantes, para priorizar la búsqueda de las causas de la anomalía.

Para la fusión jerárquica de los datos, se han estudiado cuatro escenarios distintos. El primer caso de estudio (**H1**) tiene dos niveles o capas, mientras que el resto (**H2**, **H3** y **H4**) tiene tres capas. En los resultados obtenidos, todos los escenarios excepto **H4** presentan una capacidad de detección de anomalías comparable a la obtenida para la fusión estándar en el conjunto de datos UGR'16. **H4** presenta buenos resultados, pero su rendimiento es menor comparado con el resto de opciones. Dado que **H2**, **H3** y **H4** tienen varias capas, y que al probar otras alternativas de fusión como balanceo de ramas o concatenación de estadísticos en todas las capas los resultados no han sido determinantes, no podemos recomendar una configuración entre estas tres alternativas. En cambio, **H1** (dos capas, concatenando estadísticos) asegura un rendimiento que es comparable con las fusiones estándar de los datos.

El diagnóstico en la fusión jerárquica comienza en la capa superior, ayudando a los operadores a priorizar la búsqueda de la causa de la anomalía, permitiendo localizar la fuente de la misma. En la capa inferior (y a veces en la intermedia), el diagnóstico ayuda a descubrir las causas de la anomalía identificando las características más relevantes implicadas en dicha anomalía.

La principal ventaja que aporta la fusión jerárquica es que reduce el volumen de datos que se monitoriza. Por otra parte, ayuda a mantener la privacidad, ya que los modelos en los capas superiores de la jerarquía se pueden construir sin necesidad de utilizar las variables originales, sino utilizando los estadísticos. Además, utilizar la fusión **C** en cada capa, permite localizar la fuente de la anomalía.

Conclusiones

El núcleo de esta tesis es MSNM, que se basa en dos pilares: por una parte, NSM, procedente del área de la Seguridad IT; por otra, MSPC, del campo de estudio de los procesos industriales. Desde que MSNM se propuso hace casi cinco años, ha habido numerosas extensiones, bien mejorando alguno de sus pasos [40, 129, 192, 195], bien incluyendo pasos adicionales que añaden nuevas funcionalidades a la propuesta original [33, 36]. Este trabajo de doctorado ha contribuido activamente en algunas de estas extensiones, tales como [36, 40]. En concreto, las dos contribuciones principales presentadas como parte de esta tesis son también extensiones MSNM:

- Un enfoque de pre-procesamiento para BMSPC o MSNM ciclo-estacionario que considera más observaciones para obtener los parámetros de pre-procesamiento: PARAMO. Este enfoque mejora la capacidad de detección de anomalías del sistema de monitorización [192].
- Un método de diagnóstico univariante para mejorar la diagnosis: U-Squared. Este método reduce la complejidad del diagnóstico, ayudando a priorizar las anomalías, que es uno de los mayores problemas de los equipos de Seguridad IT. También contribuye a resolver el problema de *smearing*, bien conocido en la diagnosis MSPC.
- Finalmente, se ha realizado la evaluación de extensiones MSNM por primera vez con datos procedentes de una red real: PARAMO [192],

U-Squared [195], así como fusión jerárquica [129] la metodología *5-steps* [36].

Adicionalmente, se ha propuesto una metodología para la comparación de métodos de diagnóstico. Esta metodología *i)* genera anomalías con diagnóstico conocido, *ii)* define una métrica para la evaluación del rendimiento de los métodos de diagnóstico, y *iii)* considera los factores que afectan la diagnosis para el diseño de experimentos. Estos requerimientos se aplican siguiendo un procedimiento de Monte Carlo, lo que proporciona resultados de baja incertidumbre que, cobinados con ANOVA, permiten comparar los métodos de diagnóstico de forma no sesgada.

Este trabajo de investigación muestra la simbiosis existente entre los procesos industriales y la seguridad en redes, introduciendo mejoras que son de interés para ambas áreas y que abren nuevas líneas de investigación, explorando la sinergia entre MSPC y MSNM.