1290

UNIVERSIDADE Đ
COIMBRA

Konstantia Barmpatsalou

Uma Integração de Métodos de Aquisição Forense em Tempo Real nos Sistemas PPDR da Próxima Geração

ForEmSys: An Integration of Live Forensic Acquisition Methods in Next Generation PPDR Systems

Tese no âmbito do Programa de Doutoramento em Ciências e Tecnologias da Informação orientada pelo Professor Doutor Edmundo Heitor da Silva Monteiro e pelo Professor Doutor Paulo Alexandre Ferreira Simões e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Dezembro de 2020

Faculdade de Ciências e Tecnologia
da Universidade de Coimbra.

UMA INTEGRAÇÃO DE MÉTODOS DE AQUISIÇÃO FORENSE EM TEMPO REAL NOS SISTEMAS PPDR DA PRÓXIMA GERAÇÃO

FOREMSYS: AN INTEGRATION OF LIVE FORENSIC ACQUISITION METHODS IN NEXT GENERATION PPDR SYSTEMS

Konstantia Barmpatsalou

Tese no âmbito do Programa de Doutoramento em Ciências e Tecnologias da Informação, orientada pelo Prof. Dr. Edmundo Heitor da Silva Monteiro e pelo Prof. Dr. Paulo Alexandre Ferreira Simões, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Dezembro de 2020

1 2 9 0

UNIVERSIDADE Ð
COIMBRA

This work was partially funded by the Centro 2020 Mobitrust Project (reference CENTRO-01-0247-FEDER-003343), the QREN ICIS Project (Intelligent Computing in the Internet of Services – CENTRO-07-0224-FEDER-002003) and the SALUS FP7 Project (Security and interoperability in next generation PPDR communication infrastructures).

There is a crack in everything. That’s how the light goes in.
— Leonard Cohen

Acknowledgements

This is the beginning of the end of this fantastic 5-year-old journey. The following lines are dedicated to all of you, who are a part of this thesis.

Firstly, I would like to thank my supervisors. Edmundo Monteiro, for the continuous challenges and motivation to brainstorm and never stop questioning even myself and Paulo Simoes, for the practical co-ordination and realistic overview of this effort. They are indeed the perfect supervising balance. I would also like to thank my colleague and co-author Tiago Cruz for the countless hours of work on corrections and guidelines, and Bruno Sousa for the guidance in the PPDR world.

Moreover, a special section is dedicated to the teams of the SALUS and Mobitrust projects for the scientific contributions, discussions and funding provided for the development and conclusion of the current thesis.

This whole thesis and PhD are dedicated to my late dad Byron, who inspired me not to be afraid of the world, but keep fighting instead. Also to my mum Veta, who manages to be the sweetest and most practical person in the whole world at the same time. When I told my grandmother Eleni that I was going to be a detective, she was scared of me getting killed and advised me to become a University professor instead. With this PhD thesis, both our goals are partially achieved!

Special thanks to my MSc advisor, Prof. Georgios Kambourakis for all the trust and profound professional support since 2012, when the foundations of this effort started forming.

I would also like to thank Carlos Cortinhas and my cousins Eleni and Giannis Karamouzis for the endless patience and support over the course of these 5 years. Last but not least, a big thank you to the whole LCT family, who embraced me since day 1 and all my friends in Greece, Portugal and other parts of the world just for being who they are and keeping on with a smile every day.

Resumo

Os dispositivos móveis substituíram os computadores pessoais e portáteis em muitos aspectos da rotina diária das pessoas. Na practica, eles transformaram-se em impressões digitais que carregam uma quantidade crítica de informações pessoais, que variam desde conteúdo multimedia e registos de comunicação, a geolocalização e dados de transações eletrônicas. No entanto, o uso de dispositivos móveis não se limita às interacções pessoais de um indivíduo. Os dispositivos móveis podem constituir partes de redes de comunicação corporativas ou dedicadas.

As redes corporativas e da emergência como os sistemas de Proteção Pública e Mitigação de Desastres (PPDR), exigem o estabelecimento de um ambiente altamente seguro, para proteger vários bens críticos. Além disso, organizações como a Polícia Judiciária acedem dados de dispositivos móveis de terceiras entidades como provas para investigações criminais.

A aquisição e análise forense móvel têm um papel crucial tanto na proteção de um ambiente PPDR contra ataques intencionais ou uso indevido dos utilizadores, como na condução de uma investigação criminal robusta. Esta tese estuda o papel da aquisição e análise forense para sistemas PPDR, introduzindo uma metodologia para perfis digitais automatizados e identificação de padrões suspeitos a partir de dados e metadados de dispositivos móveis.

Três técnicas de computação inteligente, nomeadamente Fuzzy Systems, Redes Neuronais (RNs) e Adaptive Neuro-Fuzzy Inference System (ANFIS) são usadas para construir perfis criminais e identificar padrões suspeitos em dados e metadados provenientes de chamadas e SMS para três cenários de casos de uso diferentes. Mais especificamente, os Sistemas Fuzzy servíram como prova de conceito para detectar a deserção de agentes PPDR realizada por SMS. Um cenário mais complexo envolveu o uso de RNs e ANFIS, que foram empregados como meio de identificação de padrões suspeitos de chamadas e SMS para casos de cyberbullying e de tráfico de droga.

Os resultados que foram produzidos durante todas as fases experimentais foram bastante satisfatórios e foram comparados para definir a técnica mais apropriada para a identificação de padrões suspeitos.

Palavras-chave: Segurança de Informação; Proteção Pública e Mitigação de Desastres; Forense Móvel; Análise Forense de Dados Móveis; Redes Neuronais; Sistemas Fuzzy; Perfil Criminal Digital

Abstract

Mobile devices have substituted desktop and portable computers in many aspects of people's everyday routine. Practically, they have become digital fingerprints that carry a critical amount of personal information, varying from multimedia and communication logs to geolocation and electronic transaction data. Moreover, the usage of mobile devices is not limited to an individual's personal interactions. The aforementioned devices may also constitute parts of corporate or dedicated communication networks.

Enterprise and first-responder communication networks, such as Public Protection and Disaster Relief (PPDR) systems require the establishment of a highly secured environment, in order to protect various critical assets. Moreover, services such as law enforcement may need to access third-party mobile device data as evidence for criminal investigations.

Mobile forensic acquisition and analysis play a crucial role towards both the protection of a PPDR environment against intentional attacks or potential user misuse and the conduction of a robust criminal investigation. The current thesis studies the role of forensic analysis in use cases related to law enforcement investigations by introducing a methodology for automated digital profiling and suspicious pattern identification from mobile device data and metadata.

Three intelligent computation techniques, namely Fuzzy Systems, Neural Networks (NNs) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) are used for constructing criminal profiles and identifying suspicious patterns in calls and SMS evidence data and metadata for three different use case scenarios. More specifically, Fuzzy Systems served as proof-of-concept for detecting PPDR officers' defection performed by SMS. A more complex scenario for call and SMS suspicious pattern identification of cyberbullying and low-level drug dealing cases was documented with the use of NNs and ANFIS.

Keywords: Information Security; Public Protection and Disaster Relief; Mobile Forensics; Mobile Forensic Data Analysis; Neural Networks; Fuzzy Systems; Digital Criminal Profiling

Foreword

The work detailed in this thesis was accomplished at the Laboratory of Communication and Telematics (LCT) of the Centre for Informatics and Systems of the University of Coimbra (CISUC), within the context of the following projects:

Project SALUS - Security and interoperability in next generation PPDR communication infrastructures - ID: 313296. Project within the European 7th framework program, from September 2013 with the duration of three years, aiming to design, implement and evaluate a next generation communication systems for Public Protection and Disaster Relief (PPDR) agencies, supported by network operators, research institutions and industry, which will provide security, privacy, seamless mobility, Quality of Service (QoS) and reliability support for mission-critical Personal Mobile Radio (PMR) voice and broadband data services.

Project MobiTRUST - Mobitrust is focused on the security of mobile terminals for PPDR use cases. It corresponds to the national component of the European Eureka/Cluster for Application and Technology Research in Europe on NanoElectronics (CATRENE) MobiTrust project (CA208). It aims to enhance the security and privacy levels of mobile platforms, by accomplishing a high awareness level in Command and Control Centres (CCCs), and establishing trusted execution environments with advanced authentication and encryption mechanisms. Moreover, concepts such as online and offline forensic data acquisition and analysis, secure integration with sensors and peripherals and Bring Your Own Device (BYOD) paradigm support are incorporated into respective use cases.

The outcome of the design, experiments, and assessments of several mechanisms on the course of this work resulted in the following publications:

Journal publications:

- **Konstantia Barmpatsalou**, Tiago Cruz, Edmundo Monteiro, and Paulo Simoes. 2018. Mobile Forensic Data Analysis: Suspicious Pattern Detection in Mobile Evidence. *IEEE Access*, 6, 59705-59727. DOI: 10.1109/ACCESS.2018.2875068. Impact factor: 3.55
- **Konstantia Barmpatsalou**, Tiago Cruz, Edmundo Monteiro, and Paulo Simoes. 2018. Current and Future Trends in Mobile Device Forensics. *ACM Computing Surveys* 51, 3, 1–31. DOI: 10.1145/3177847. Impact factor: 6.74

Conference publications:

- **Konstantia Barmpatsalou**, Tiago Cruz, Edmundo Monteiro, and Paulo Simoes. 2017. Fuzzy system-based suspicious pattern detection in mobile forensic evidence. In *Proceedings of the 9th EAI International Conference on Digital Forensics and Cyber Crime*, EAI, Prague, Czech Republic.DOI: 10.1007/978-3-319-73697-6_8
- **Konstantia Barmpatsalou**, Bruno Sousa, Edmundo Monteiro, and Paulo Simoes. 2015. Mobile Forensics for PPDR Communications: How and why?. In *Proceedings of the 10th International Conference on Cyber Warfare and Security (ICCWS2015).* Kruger National Park, South Africa, 30-38. ISBN: 978-1-910309-96-4
- **Konstantia Barmpatsalou**, Edmundo Monteiro, and Paulo Simoes. 2014. Mobile Forensics: Evidence Collection and Malicious Activity Identification in PPDR Systems. In *Proceedings of the International Conference in Information Security and Digital Forensics (ISDF2014).* SDIWC, Thessaloniki, Greece, 42-48. ISBN: 978-1-941968-03-1

Book chapter publications:

- **Konstantia Barmpatsalou**, Tiago Cruz, Edmundo Monteiro, and Paulo Simoes. 2017. From fuzziness to criminal investigation: An inference system for Mobile Forensics. In *Intrusion Detection and Prevention for Mobile Ecosystems*, G. Kambourakis, A. Shabtai, C. Kolias, D. Damopoulos (Eds.). CRC Press, 117-133. DOI: 10.1201/9781315305837-6

Other publications:

- Hugo Marques, Luis Pereira, Jonathan Rodriguez, Georgios Mantas, Bruno Sousa, Hugo Fonseca, Luis Cordeiro, David Palma, **Konstantia Barmpatsalou**, Paulo Simoes, Edmundo Monteiro, Andy Nyanyo, Peter Wickson, Bert Bouwers, Branko Kolundzija, Dragan Olcan, Daniel Zerbib, Jerome Brouet, Philippe Lasserre, Panagiotis Galiotos, Theofilos Chrysikos, David Jelenc, Jernej Kos, Denis Trcek, Alexandros Ladas, Nuwan Weerasinghe, Olayinka Adigun, Christos Politis, and Wilmuth Muller. (2015). Next-Generation Communication Systems for PPDR: the SALUS Perspective. In *Wireless Public Safety Networks 1.* Elsevier, 49-93. DOI: 10.1016/B978-1-78548-022-5.50003-0

Contents

Acknowledgements **ix**

Resumo **x**

Abstract **xi**

Foreword **xii**

List of Figures **xviii**

List of Algorithms **xix**

List of Tables **xxi**

Abbreviations and Acronyms **xxi**

1 Introduction **1**

1.1 Motivation and Problem Statement 1

1.2 Objectives and Contributions 3

1.3 Outline of the Thesis 4

2 General Background and Methodology **6**

2.1 Introduction 6

2.2 Mobile Forensics Contextualization 7

2.2.1 Mobile Forensics Principles 10

2.2.2 Cloud Computing and Mobile Forensics 15

2.2.3 Mobile Forensic Practices 17

2.3 Recent Advances in Mobile Forensics 19

2.3.1 File Acquisition and Data Integrity 20

2.3.2 Identification of Malicious Activity and Malware Analysis 25

2.3.3 Evidence Reconstruction and Presentation 27

2.3.4 Evidence Parsing 28

2.3.5 Automated Classification and Analysis of User and Application Behaviour 30

2.4 Mobile Forensic Data Analysis and Digital Criminal Profiling . 32

2.4.1 Digital Criminal Profiling Principles 32

2.4.2 Related Work 35

2.5 Proposed Methodology 39

2.6 Summary of the Chapter 42

3 Fuzzy Systems for Suspicious Pattern Identification 43
3.1 Fuzzy Systems . . . 43
3.1.1 Fuzzy Membership Functions . . . 44
3.1.2 The Fuzzy Inference Mechanism . . . 49
3.1.3 Fuzzy System Types . . . 53
3.2 Use Case Definition . . . 54
3.2.1 Rule Inference . . . 55
3.3 Dataset Selection . . . 56
3.4 Pre-Processing . . . 57
3.5 Ground Truth Generation . . . 58
3.6 Evaluation . . . 59
3.7 Summary of the Chapter . . . 61

4 Neural Networks and ANFIS for Suspicious Pattern Identification 62
4.1 Neural Networks . . . 63
4.1.1 Neural Networks Architecture . . . 65
4.1.2 Neural Network training . . . 68
4.1.3 Backpropagation Algorithms . . . 69
4.2 Neuro-Fuzzy Systems . . . 77
4.2.1 ANFIS . . . 78
4.3 Methodology . . . 81
4.3.1 Use Case Definition . . . 81
4.3.2 Datasets . . . 82
4.3.3 Pre-processing . . . 83
4.3.4 Common Attributes . . . 83
4.3.5 Call-exclusive Attributes . . . 84
4.3.6 SMS-exclusive Attributes . . . 85
4.3.7 Ground Truth Generation . . . 85
4.4 Neural Networks and ANFIS . . . 89
4.4.1 Plain and Pattern Recognition Backpropagation Perceptron Configuration . . . 89
4.4.2 Evaluation . . . 93
4.4.3 Testing on Unknown Data . . . 93
4.5 Summary of the Chapter . . . 95

5 Results 96
5.1 Fuzzy Systems Results . . . 96
5.2 Neural Networks and ANFIS Results . . . 98
5.2.1 CDA Dataset Results . . . 99
5.2.2 Best Performance Perceptron and Algorithm Selection . 107
5.2.3 Testing on Unknown Data . . . 108
5.3 Summary of the Chapter . . . 111

6 Conclusions and Future Work 113
6.1 Synthesis of the Thesis . . . 113
6.2 Contributions . . . 114
6.3 Discussion and Future Work . . . 114

Bibliography **117**

List of Figures

2.1 MF contextualization 8
2.2 Mobile Forensics investigative process model, extended from Ayers et al. 2014 [Ayers et al., 2014] 10
2.3 Detailed acquisition phase, introduced by Barmpatsalou et al. [Barmpatsalou et al., 2018a] 12
2.4 Inductive analysis, adapted from Godwin [Godwin, 2012] . . . 34

3.1 Triangular membership function 45
3.2 Trapezoidal membership function 45
3.3 Bell membership function 46
3.4 Gauss membership function 47
3.5 Gauss2 membership function 48
3.6 Fuzzy System architecture, adapted from Kumar and Verma [Kumar and Verma, 2015] and Abraham [Abraham, 2005] 48
3.7 Fuzzification for different membership function types, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 49
3.8 Max-membership defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 50
3.9 Centre of gravity defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 51
3.10 Weighted average defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 52
3.11 Mean-max membership defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 52

4.1 Neural node sample, adapted from Siddique and Adeli [Siddique and Adeli, 2013] 64
4.2 Single-layer feedforward NN, adapted from da Silva et al. [da Silva et al., 2017] 66
4.3 Multi-layer feedforward NN, adapted from da Silva et al. [da Silva et al., 2017] 66
4.4 Recurrent NN, adapted from da Silva et al. [da Silva et al., 2017] 67
4.5 Kohonen (mesh) NN, adapted from da Silva et al. [da Silva et al., 2017] 67
4.6 Cooperative NFS, adapted from Vieira et al. [Vieira et al., 2004] 77
4.7 Concurrent NFS, adapted from Vieira et al. [Vieira et al., 2004] 78
4.8 ANFIS, adapted from Jang [Jang, 1993] and Suparta and Alhasa [Suparta and Alhasa, 2016] 79
4.9 Generic plain and pattern recognition perceptron architecture . 90
4.10 Average MSE per number of neurons - BP calls 91

4.11 Average MSE per number of neurons - BP SMS 91
4.12 Average MSE per number of neurons - BPN SMS 92
4.13 Average MSE per number of neurons - BPN calls 92
4.14 ADAET Functionality, extended from Spreitzenbarth and Uhrmann [Spreitzenbarth and Uhrmann, 2015] 94

5.1 Performance histogram for the plain backpropagation perceptron - Cyberbullying: Calls 100
5.2 Performance histogram for the plain backpropagation perceptron - Cyberbullying: SMS 101
5.3 Performance histogram for the plain backpropagation perceptron - Drug Dealing: Calls 104
5.4 Performance histogram for the plain backpropagation perceptron - Drug Dealing: SMS 105
5.5 Average performance per training algorithm 112

List of Algorithms

4.1 Calls preprocessing 86
4.2 SMS preprocessing 87

List of Tables

2.1 Acquired data types, extended from Barmpatsalou et al. [Barmpatsalou et al., 2013] . . . 12
2.2 Digital criminal profiling papers' characteristics . . . 38
2.3 Methodology Compilation . . . 39
2.4 Methodology Phases . . . 40

3.1 Fuzzy variable ranges . . . 57
3.2 Fuzzy suspiciousness values, adapted from Barmpatsalou et al. [Barmpatsalou et al., 2018b] . . . 59
3.3 Ground truth generation per rule . . . 60

4.1 Time quantification . . . 84
4.2 Country code and mobility quantification . . . 85
4.3 Fuzzy Suspiciousness Values, adapted from Barmpatsalou et al. [Barmpatsalou et al., 2018b] . . . 88
4.4 Output Transformations for the Pattern Recognition Perceptron 88
4.5 CDA GT Pattern Distribution . . . 89

5.1 Evaluation metrics per membership function for the SMS Dev. 1 dataset . . . 97
5.2 Evaluation metrics per membership function for the SMS Dev. 2 dataset . . . 97
5.3 Evaluation metrics per membership function for the SMS Dev. 3 dataset . . . 98
5.4 Plain Backpropagation Perceptron Performance - Cyberbullying: Calls . . . 100
5.5 Plain Backpropagation Perceptron Performance - Cyberbullying: SMS . . . 100
5.6 Backpropagation Pattern Recognition Perceptron Performance - Cyberbullying: Calls . . . 101
5.7 Backpropagation Pattern Recognition Perceptron Performance - Cyberbullying: SMS . . . 102
5.8 ANFIS Performance - Cyberbullying: Calls . . . 102
5.9 ANFIS Performance - Cyberbullying: SMS . . . 103
5.10 Plain Backpropagation Perceptron Performance - Drug Dealing: Calls . . . 104
5.11 Plain Backpropagation Perceptron Performance - Drug Dealing: SMS . . . 104
5.12 Backpropagation Pattern Recognition Perceptron Performance - Drug Dealing: Calls . . . 105

5.13 Backpropagation Pattern Recognition Perceptron Performance - Drug Dealing: SMS 106
5.14 ANFIS Performance - Drug Dealing: Calls 106
5.15 ANFIS Performance - Drug Dealing: SMS 107
5.16 Samsung Device GT Pattern Distribution 109
5.17 Samsung Device Backpropagation Perceptron Performance - Cyberbullying: Calls 109
5.18 Samsung Device Backpropagation Perceptron Performance - Cyberbullying: SMS 110
5.19 Samsung Device Backpropagation Perceptron Performance - Drug Dealing: Calls 110
5.20 Samsung Device Backpropagation Perceptron Performance - Drug Dealing: SMS 111

Abbreviations and Acronyms

ADAET Android Data Acquisition and Examination Tool

ADALINE ADAptive LInear NEuron

ADB Android Debug Bridge

AF Audio Forensics

AFDA Android Forensic Data Analyzer

ANFIS Adaptive Neuro-Fuzzy Inference System

API Application Programming Interface

AProcDump Android Process Dumper

ARM Advanced RISC Machine

AUC Area Under the Curve

BEA Behavioural Evidence Analysis

BYOD Bring Your Own Device

CDA Cambridge Device Analyzer

CDMA Code Division Multiple Access

CF Computer Forensics

CFReDS Computer Forensic Reference Data Sets

CRISP-DM CRoss-Industry Standard Process–Data Mining

CSV Comma Separated Value

DBaaS Database-as-a-Service

DBF DataBase Forensics

DENFIS Dynamic Evolving Neural-Fuzzy Inference System

DF Digital Forensics

DFIAC Digital ForensIcs Analysis Cycle

EU European Union

FALCON Fuzzy Adaptive Learning COntrol Network

FINEST Fuzzy Inference and NEural network in fuzzy inference SofTware

FN False Negative

FP False Positive

FPR False Positive Rate

FROST Forensic Recovery of Scrambled Telephones

FTK Forensic ToolKit

FUN FUzzy Network

GARIC Generalized Approximate Reasoning-based Intelligence Control

GPETD Greek Police Escort Teams Department

GPRS General Packet Radio Service

GPS Global Positioning System

GSM Global System for Mobile communications

GUI Graphical User Interface

HDD Hard Disk Drive

HIDS Host-based Intrusion Detection System

IDFPM Integrated Digital Forensics Process Model

IDS Intrusion Detection System

IEC International Electrotechnical Commission

IMEI International Mobile Equipment Identity

IMSI International Mobile Subscriber Identity

IM Instant Message/ing

IoT Internet of Things

ISO International Organization for Standardization

IT Information Technology

JTAG Joined-Test Action Group

kNN k-Nearest Neighbour

LiME Linux Memory Extractor

LSE Least Square Estimate

LTE Long-Term Evolution

MDM Mobile Device Management

MDS MultiDimensional Scaling

MIMO Multi-Input Multi-Output

MISO Multi-Input Single-Output

MF Mobile Forensics

MFDA Mobile Forensic Data Analysis

MSE Mean Square Error

MO Modus Operandi

NAND Negative AND

NB Naive Bayes

NEFCON NEuro-Fuzzy CONtrol

NF Network Forensics

NFAT Network Forensics Analysis Tool

NFI Netherlands Forensic Institute

NFN Fuzzy Neural Network

NFS Neuro-Fuzzy System

NIDS Network Intrusion Detection System

NN Neural Network

OS Operating System

PAN Personal Area Network

PFSM Probabilistic Finite State Machine

PIN Personal Identification Number

PMR Private Mobile Radio

PPDR Public Protection and Disaster Relief

RAM Random Access Memory

REST REpresentational STate

RFC Random Forest Classification

SIM Subscriber Identity Module

SMS Short Message Service

SN Social Network

SONFIN Self COnstructing Neural Fuzzy Inference Network

SQL Structured Query Language

SSH Secure SHell

SVM Support Vector Machine

TCO Total Cost of Ownership

TETRA TErrestrial Trunked RAdio

TN True Negative

TP True Positive

UFED Universal Forensic Extraction Device

UNODC United Nations Office on Drugs and Crime

URL Uniform Resource Locator

VF Video Forensics

WSN Wireless Sensor Network

XML eXtensive Markup Language

Chapter 1

Introduction

> Does the walker choose the path, or the path the walker?
>
> *(Garth Nix)*

The current thesis aims to describe the role of mobile forensic analysis within the environment of mobile communication systems, as applied in criminal investigation and law enforcement. The application and evolution of the aforementioned fields is researched and the digital profiling and suspicious pattern identification issues are presented and addressed. In the rest of the chapter, the motivation and objectives of the thesis are presented and its outline is provided.

1.1 Motivation and Problem Statement

During the last decade, smartphones have shown increased computational and networking capabilities. End-users are enjoying improved quality of communications, especially concerning data transfer services [Pande, 2013] in commercial, enterprise and dedicated, PPDR systems. PPDR infrastructures "are used by agencies and organizations dealing with the maintenance of law and order, the protection of life and property and with emergencies" [Jamieson, 2004].

Law enforcement agencies are a subcategory of PPDR systems that is more affected by the evolution of mobile devices in several levels. Firstly, mobile devices are more prone to various types of malicious activity against the participating devices and the networks themselves. The aforementioned types of malicious activity may derive either from external offenders or from internal intentional or unintentional misuse of mobile devices.

Apart from threats against their infrastructures, law enforcement agencies have to investigate crimes that involve the use of mobile devices and vary from malware propagation and fraud to classic crime types where mobile devices are involved as the perpetrators' means of communication and their digital fingerprint can provide evidence useful for their framing.

Mobile Forensic Data Analysis (MFDA) may play an important role among those security solutions, assisting on the detection and profiling of malicious attacks against law enforcement systems, the gathering of legal evidence and their investigation in search of suspicious data and metadata patterns. The current thesis presents the contribution of MFDA in criminal investigation within the context of a law enforcement system, while focusing more on suspicious behaviour identification from mobile evidence. The choice upon the thesis' scope was made after the following research question was raised.

Why is behaviour-based criminal investigation in such a need of scientific contributions, when similar disciplines relying on software-based investigation, such as malware identification or intrusion detection have advanced in such a high pace?

The answer lies in the fact that software- or machine-based activity, such as malware or botnet propagation has a more easily defined and therefore more predictable fingerprint. On the contrary, when actions performed on a device are related to human behaviour, the number of combinations and subsequent outcomes increases significantly. As a result, the creation of a behavioural model is a rather complex task. Moreover, human behaviour is mainly a qualitative variable. Matching and pointing qualitative characteristics to quantitative representations is not a straightforward, but a rather intuitive process and requires lots of trial-and-error attempts.

This is also one of the reasons why Mobile Forensic (MF) research has "fallen into the trap of focusing almost exclusively on the collection of data and has paid very little attention to the examination and analysis phases" [Rogers, 2016]. As a result, the need for the implementation of intelligent solutions that will cast off the burden of manual investigations is immediate and crucial [Rogers, 2003], [Kasiaras et al., 2014], [Ntantogian et al., 2014], [Barmpatsalou et al., 2018a].

The role of data and metadata forensic investigation is double. Firstly, it is a failsafe mechanism in case direct access to evidence does not succeed in producing a concrete outcome due to anti-forensic scenarios, such as data cascading or deliberate data alteration, such as encoded verbal communication between criminals. Secondly, it can become a means of off-loading investigators' tasks, serving as a triage mechanism for potentially suspicious user behavioural patterns before or after a hands-on investigation.

The conclusions reached in the previous paragraphs lead to a research-question follow-up:

How to perform suspicious pattern identification based on user behavioural data from mobile devices?

The answer is derived from the observation of the differences between software-

and human-generated activity. Solutions for software-based threats use a binary classification methodology so as to define if a set of attributes is either culpable of an accusation or not. However, in traditional criminal cases that involve the use of mobile devices as means of communication, binary classification is not an efficient approach.

There is a probability that binary classification may lead to excessive false positive or false negative production, states that reduce the overall quality of the mechanism. However, this side effect can be avoided if the model's output has a higher number of output states, that determine different scales of belonging to a particular condition (true or false). A multiclass output space allows for better observation and understanding of intermediate values and thus, it is able to result in a more accurate decision making procedure.

Intelligent Computation methods, such as Fuzzy Systems, Neural Networks (NNs) and Neuro-Fuzzy Systems specialize in producing results within a multiclass output space. This fact renders them the most appropriate candidates for resolving issues in an uncertain universe, such as the suspicious pattern identification of mobile forensic evidence for pre-identified crime types, based on the observation of the criminals' behaviour and interaction with the devices.

1.2 Objectives and Contributions

The objectives of the current thesis are the contextualization of mobile forensic acquisition and analysis in PPDR systems and the provision of a methodology that addresses an important research gap, the lack of automatic digital criminal profiling and suspicious pattern identification of forensic evidence.

More precisely, the thesis presents a methodology that performs forensic analysis on mobile call and SMS data and metadata series of attributes, aiming to identify suspicious patterns that constitute a criminal Modus Operandi (MO). Three types of intelligent computation techniques, namely Fuzzy Systems, Neural Networks (NNs) and the Adaptive Neuro-Fuzzy Inference System (ANFIS)are used as evaluation mechanisms in a hybrid dataset consisting of actual and simulated mobile device evidence. Different configurations for each technique are tested in a series of experiments and the best performing technique is selected.

The specific goals of this thesis are the following:

Goal 1 - To highlight the importance of MFDA within a dedicated environment;

Goal 2 - To create a methodology that combines automatic criminal profiling and pattern identification from evidence data and metadata;

Goal 3 - To provide substantial proof of concept for the aforementioned methodology by applying it to different types of intelligent systems and evaluating them accordingly.

The aforementioned goals are the springboard to the following contributions:

Contribution 1 - A Methodology for Suspicious Pattern Identification
Based on an extensive research on the relevant MFDA literature, we propose a new digital criminal profiling and suspicious pattern identification methodology that retains the evolutionary characteristics of its predecessors, such as the continuous interaction between the profiling characteristics and the new input data, but is also capable of assigning suspiciousness values to different data and metadata patterns.

Contribution 2 - Suspicious Pattern Identification with Fuzzy Systems
A proof of concept for the proposed methodology for a small use case scenario, aiming to profile the MO of PPDR officers defecting to the rioters' side by examining their sent SMS and to identify the respective suspicious patterns. Mamdani Fuzzy Systems with different configurations are used for the identification procedure and their performance is evaluated.

Contribution 3 - Suspicious Pattern Identification with NNs and ANFIS
Two complex use case scenarios, involving the profiling and identification of call and SMS patterns for cyberbullies and low-lever drug dealers are examined with the proposed methodology. NNs and ANFIS are configured and employed as evaluation tools. The performance of different setup is then measured, the prevailing solution is selected and tested anew on previously unknown data for the system.

The following subsection presents the structure of the current thesis

1.3 Outline of the Thesis

The rest of the thesis is organized in the following manner.

Chapter 2 – General Background and Methodology
The chapter performs a State-of-the-Art (SoA) analysis of the MF discipline, contextualizes its role within the PPDR ecosystem and identifies the potential research gaps. It also performs a bibliographic analysis on the concepts of MFDA and digital criminal profiling. Once the related work in the field is presented, the methodology used in the current thesis is elaborated.

Chapter 3 – Fuzzy Systems for Suspicious Pattern Identification
The chapter presents the first part of the methodology application, that involved the use of Fuzzy Systems and served as a proof of concept for suspicious pattern identification in mobile forensic evidence. An introduction on the basic Fuzzy Systems concepts is performed and the

rest of the chapter is concerned with applying and adapting the proposed methodology to the PPDR officers' infiltration-by-SMS use case.

Chapter 4 – Neural Networks and ANFIS for Suspicious Pattern Identification

The current chapter is split in two parts. The first performs an introduction to fundamental concepts of NNs and the ANFIS, whereas the second applies and adapts the methodology to the cyberbullying and drug dealing use cases.

Chapter 5 – Results

Provides the results from the experiments performed both with Fuzzy Systems, NNs and ANFIS and presents the performance ratings per use case and technique variation utilized.

Chapter 6 – Conclusions and Future Work

Enumerates the conclusions that emerged from the current research, summarizes the thesis and proposes future contributions for the evolution of the proposed methodology.

Chapter 2

General Background and Methodology

Science gave us forensics.
Law gave us crime.

(Mokokoma Mokhonoana)

The current chapter aims to provide an in-depth analysis on how the discipline of Mobile Forensics (MF) has evolved overtime. Once the analysis of the research background and the enumeration of potential challenges are completed, the chapter covers the advances of Mobile Forensic Data Analysis (MFDA), while focusing on behavior-based suspicious pattern identification principles. A considerable part of this chapter has already been published in the form of a paper [Barmpatsalou et al., 2018a].

2.1 Introduction

The increased involvement of electronic devices in criminal actions “has led to the development of Digital Forensics (DF)” [Palmer, 2001], a discipline concerning collection, investigation, and presentation of evidence in an accepted manner upon court. However, the term *digital* incorporates many categories that cannot be regarded as a whole, and therefore, they require further classification. Some of the DF sub-disciplines encountered throughout literature encompass aspects such as Computer (CF), Network (NF), Database (DBF), Audio (AF), Video (VF) and Mobile Forensics (MF) [Shanableh, 2013].

Despite the similar functionalities of mobile devices and computers, they cannot be handled in the same way during a criminal investigation. Substantial differences in terms of hardware, software, power consumption and overall mobility make them unsuitable for classification under the CF category. As a result, the MF discipline was formulated so as to incorporate the criminal investigation of different types of mobile devices (handsets, tablets and more recently wearable devices). Fundamentally, MF “is the process of gathering evidence of some type of incident or crime that has involved mobile devices" [D' Orazio et al., 2014]. More precisely, it is in charge of the whole routine of "gathering, retrieving, identifying, storing and documenting" [Marturana et al., 2011] evidence from mobile communication devices.

Mobile device operation has its own specific constraints, constituting a compromise between processing power usage, storage capabilities and portability/autonomy. The progressive balancing and/or offload of computing resources to external entities has provided a solution to cope with device shortcomings, thus creating an intersection between the mobility concept and the Cloud. While this strategy provides a solution for dealing with device energy, storage and processing power trade-offs, it also brings new challenges, as Cloud Systems can potentially host relevant evidence.

For many, Cloud Computing is the future of mobility. In a recent survey by the Right Scale company [RightScale, 2016], 95% of the surveyed organizations have adopted a private, public or hybrid Cloud strategy. In the same survey, security on the Cloud is ranked second in the list of the most precarious issues in need of improvement. Such a concern is rather realistic: since Cloud Services cope with increased amounts of sensitive data, they are expected to become a preferred target of criminal activity. This creates a whole new perspective for DF, beyond the self-contained device approach. Moreover, it generates new requirements for the performance of robust investigations.

MF is based on the premise that mobile devices contain important information about an individual's personal or professional activities, which are crucial pieces of evidence during an investigation. As the amount of valuable data stored in Cloud Services increases, traditional MF techniques cannot solely focus on mobile devices. Cloud Forensics addresses this gap, expanding the scope of the investigation process to the Cloud environment and encompassing Computer, Network and Mobile Forensics concepts.

The next section aims to contextualize the role of MF in infrastructures where mobile devices are interconnected in order to serve purposes varying from everyday to extended scenarios involving enterprise, law enforcement and first responders' usage.

2.2 Mobile Forensics Contextualization

Despite being tools for simplifying daily tasks, mobile devices can also be abused for criminal purposes. In this perspective, Internet-connected devices are particularly vulnerable, as they can easily become targets or even active participants, by performing attacks and spreading cyber threats. The need to investigate these events has prompted for the adoption of guidelines similar to those used for traditional forensics, in the form of DF. DF is the science of retrieving evidence out of digital devices with legally and scientifically acceptable methodologies for "preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence" [Palmer, 2001]. The aim of the aforementioned chain of actions is to provide substantial aid to forensic specialists in terms of reconstructing events and generating reports associated to the crime scene. However, the technological differences among the existing digital media led to the creation of different DF subcategories, varying from CF and NF, to AF, VF, DBF and MF. Fig. 2.1 depicts the contextualization of MF within a contemporary digital environment, which is also described in the following paragraphs.

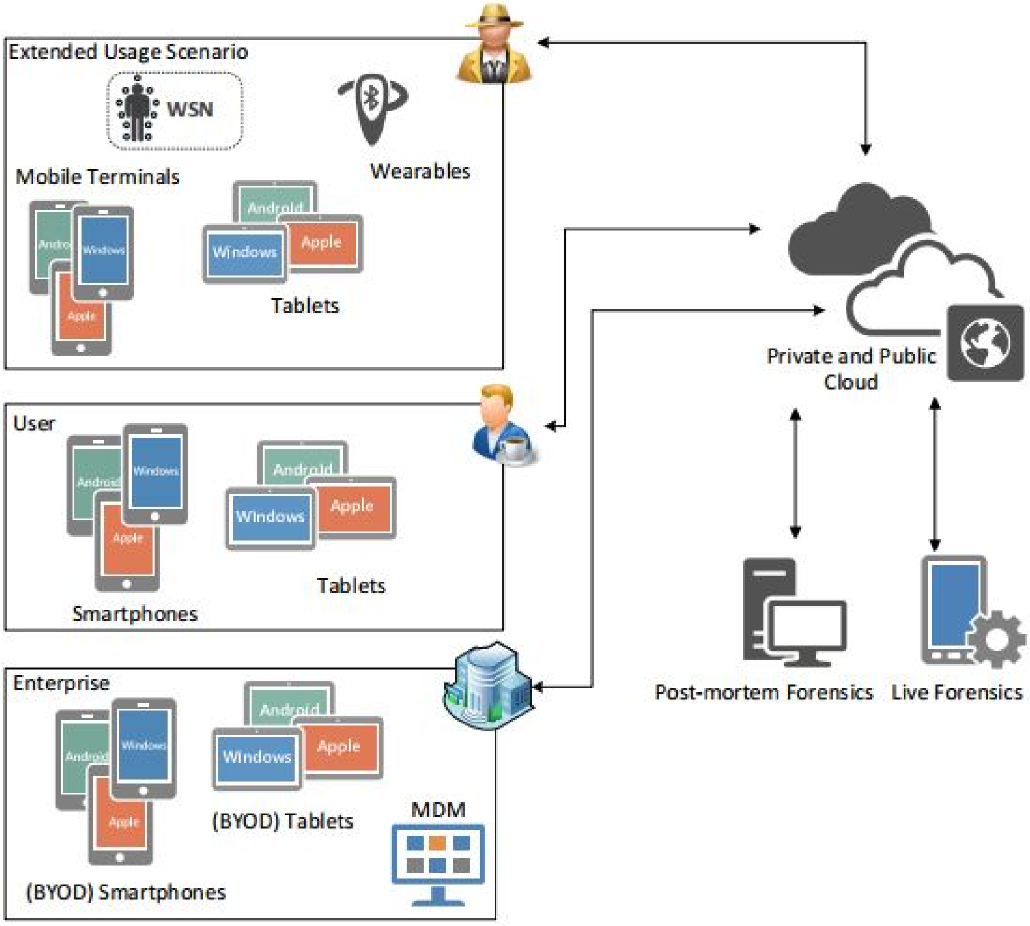

Figure 2.1: MF contextualization

MF is concerned with several aspects which are orthogonal to the mobile device ecosystem, such as usage profiles or managed asset requirements. This is a direct consequence of the pervasive role mobile devices have acquired as personal and business tools in our daily lives. A smartphone will likely reveal more details about the user's habits and behavior than a desktop or a notebook computer. Moreover, MF needs to transcend the device boundaries, encompassing the aforementioned public and private Cloud Service domains. This adds complexity and expands the boundaries of forensic investigation beyond the traditional post-mortem examination, which takes place on destroyed or powered-off devices. In such a volatile environment as the Cloud, more recent live techniques, that occur on a powered-on device or system, have proven highly efficient [Barmpatsalou et al., 2018b]. Furthermore, mobile devices often act as mediators for Personal Area Networks (PANs) (wearables), Wireless Sensor Networks (WSNs) or Internet of Things (IoT) devices. Data flows between these devices are also of potential interest for forensic purposes.

Until recently, the perception of mutual exclusivity between personal and business usage profiles deemed the need for separate devices, as it was inconceivable to use the same equipment for both roles. It was assumed that companies had no other choice than to provide their workforce with the mobile equipment required for professional usage – in order to ensure adequate control over costs, management and security. Lately, several organizations have started encouraging employees to use their own devices within the corporate environment, in an effort to reduce the Total Cost of Ownership (TCO) for mobile assets. This Bring Your Own Device (BYOD) principle implies that enterprise networks no longer consist exclusively of corporate devices. As such, Information Technology (IT) staff is prompted to "adopt more flexible and creative solutions in order to maintain a satisfactory security level, while enabling access to collaborative technologies" [Thomson, 2012].

Enterprise environments are in greater need of protection than individuals. The amount of assets to be protected and the sensitive nature of information stored and transmitted makes them a more attractive target to any sort of illegal activity. Within such environments, "Mobile Device Management (MDM)" [Souppaya and Scarfone, 2013] platforms provide organizations with the means to establish and enforce managed device policies via a dedicated platform. After enrolling in the platform and installing a MDM client application, devices start being monitored and the platform policy starts being enforced (e.g., restricting usage to corporate applications). MDM monitoring is a prerequisite, especially for BYOD users that already have a certain level of unknown interaction with the device before enrolling. This avoids exposure to untrusted content or applications that may cause irreversible damage. In this perspective, MDM helps to establish the basic security principles to fit the requirements of each organization.

Additionally, organizations that act as first responders, such as law enforcement agencies or emergency services make use of Public Protection and Disaster Relief (PPDR) systems so as to carry out a variety of tasks ensuring their robust operation. PPDR systems are critical infrastructures in great need of protection from attacks against their integrity and availability and also require secure and non-disrupted end-to-end communications. MF act as auditing mechanisms that allow for the further comprehension of actions that constitute threats to the PPDR systems and provide insights for the implementation of efficient countermeasures [Barbatsalou et al., 2015]. Particularly for law enforcement agencies, the contribution of MF is bidirectional. Not only do they fulfill the aforementioned roles, but they also serve as means of investigation for criminal cases involving mobile devices as active participants or digital fingerprints.

Considering the fact that contemporary mobile devices are becoming apt at replacing desktop and notebook computers for a variety of tasks, it could be deducted that CF-like techniques might be applied during their forensic investigation. This reasoning proves wrong, as the similarities between the two device categories are only superficial. In fact, hardware and software components have substantial differences between computers and mobile devices. As a result, different techniques have to be implemented so as to carry out a successful investigation. Nonetheless, specific smartphone components such as external SD cards can be examined effectively by classic CF methods [Hoog, 2011], but this is not enough to cover more critical parts of mobile devices, such as the flash memory. The next subsection analytically explains the principles that influence and regulate MF investigations.

2.2.1 Mobile Forensics Principles

All the aforementioned factors resulted in the birth of a separate discipline for MF, a field dedicated solely to forensic investigation in mobile devices, and which will be presented analytically in the next sections. The aspects of the investigation procedure, acquisition methods and data types will be covered in detail in the following paragraphs.

2.2.1.1 Investigation Phases in Mobile Forensics

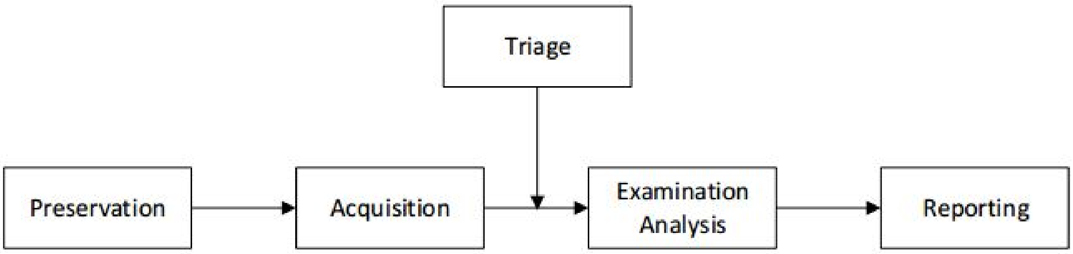

Figure 2.2: Mobile Forensics investigative process model, extended from Ayers et al. 2014 [Ayers et al., 2014]

The process model for conducting forensic investigations on mobile devices includes the following stages: “preservation, acquisition, examination/analysis and reporting of digital evidence” [Ayers et al., 2014] (see Fig. 2.2). It is a structured procedure that investigators need to follow upon device seizure. It provides guidance and recommendations for secure preservation and storage, device handling, as well as user, application and network activity tracing.

Preservation includes all the tasks first responders are responsible for. Particularly for MF, it consists of seizing and securing the mobile devices, tracking their state and ensuring that no intentional or unintentional alteration will occur to them or their contents [Raghav and Saxena, 2009]. Afterwards, during the acquisition phase, a bitwise replication or parts of the internal device memory and peripherals are extracted so as to provide the investigation material for the examination and analysis phase. Its purpose is to extract conclusions about the criminal actions by "applying established scientifically based methods to acquired evidence. Meanwhile, the examination and analysis phase should describe the content and state of the data, including the source and the potential significance" [Chen et al., 2011]. Finally, during reporting, every relevant detail or incident observed in the previous phases is completely documented, preferably in a correct chronological order.

Marturana et al. [Marturana et al., 2011] proposed enhancements for the process model, such as quantitative approaches or the inclusion of a triage stage [Rogers et al., 2006] between the acquisition and examination/analysis phases. Recently acquired data are normalized before analysis, so as to be kept relevant to the investigation needs and avoid delays caused by big amounts of raw information. However, this latter proposal is still undergoing preliminary research and is yet to be incorporated into the aforementioned MF standards as a standalone stage.

Despite the recognized significance of all the investigation process model phases, the amount of research dedicated to each part is uneven. This is due to the fact that not every stage is equally important for all fields. For example, even though data preservation is critical for the investigation itself, most of its procedures are fixed and concern notions such as chain of custody and physical security, which have already been extensively researched in the past. Moreover, the majority of preservation techniques, such as the use of Faraday cages for device isolation, require the involvement of disciplines other than Computer Science. Overall, the fields of acquisition and examination/analysis show increased research activity when compared to the other two.

The two aforementioned phases are the most related to the current thesis as well. Acquisition is fundamental for the evidence validity and thus for the failsafe conduction of the examination and analysis process. The following subsection constitutes a preamble to the acquisition phase principles, by introducing the data types that can be acquired during a MF investigation.

2.2.1.2 Acquired Data Types

Before proceeding to the MF acquisition methods presentation, it is important to allude to the evidence data types that are retrieved during an investigation, according to their types and the entity accesses them. Table 2.1 presents a taxonomy of the acquired data, as it was presented in a survey paper by Barmpatsalou et al. [Barmpatsalou et al., 2013].

The first group consists of data handled and altered strictly by Operating Systems (OSs), such as connection handlers (Global Positioning System (GPS), WiFi) and OS defaults and structural elements (IMEI, IMSI). The second group concerns data imported and edited by users, such as text messages, contact lists,

Data Category	Data Type
OS	GPS, Compass and Accelerometer Handlers, Connectivity Properties, Network Data, Installed Application Packages, OS Metadata
User	Call Logs, SMS, Instant Messages (Chat), Contacts, Multimedia Files, Browsing Data, Photographs, Videos, Office Documents, Calendar Entries, Notes, User File Metadata
Native and Cloud Application	Timestamps, Installation Data, Saved Settings, Trash, Permissions (Android), Credentials, Mobility Data, Application Metadata

Table 2.1: Acquired data types, extended from Barmpatsalou et al. [Barmpatsalou et al., 2013]

pictures and all sorts of customized application data. Data used by native and Cloud applications as background procedures, such as timestamps, installation data, mobility data and saved settings form the third category. OS, user and application metadata are also present in each of the aforementioned categories, given the importance they add to an investigation, since they provide additional information and enhance the quality of the potential findings [Ho et al., 2018]. The following subsection describes the most important concepts of the mobile forensic data acquisition process.

2.2.1.3 Mobile Forensic Acquisition Methods

Data acquisition is a popular research area within the MF discipline, mainly because its proper execution is crucial for a successful investigation. Without a successfully extracted and validated memory image or part of the file system, performed with respect to the rules of forensic soundness [Vomel, 2013], it is impossible for the rest of the procedure to take place. Figure 2.3 presents the main areas of MF acquisition from a technical point of view, as they appeared in various research papers throughout literature.

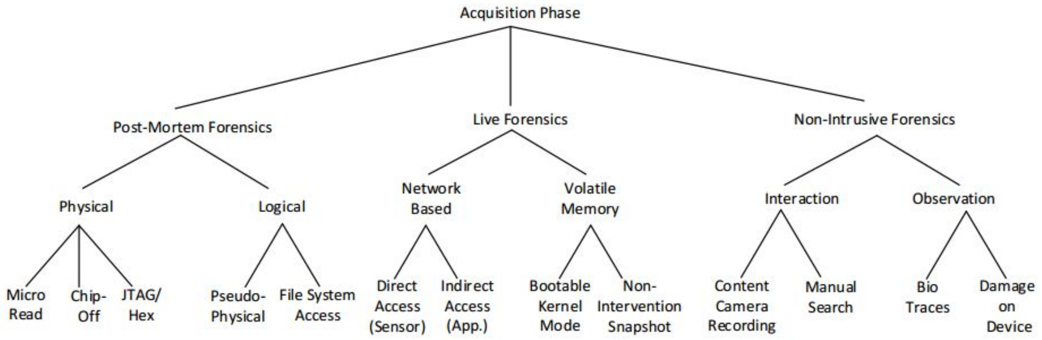

Figure 2.3: Detailed acquisition phase, introduced by Barmpatsalou et al. [Barmpatsalou et al., 2018a]

The basic acquisition methods comprise the post-mortem, live and non-intrusive forensics categories. Post-mortem, also known as dead forensics, includes physical and logical acquisition methods and takes place upon the seizure of damaged, destroyed or powered down devices, requiring a bit-by-bit copy of their memory. Acquisition takes place with devices in off-line mode (i.e., without any kind of network connectivity), so as to avoid minimal modification of its contents [Jansen and Ayers, 2007]. However, recent research [Barmpatsalou et al., 2013] reveals a trend towards alternative directions, such as the usage of boot loader modifications, which ensure the forensic soundness of the data partition, and the real-time acquisition of volatile memory contents, which are able to collect crucial evidence [Dezfouli et al., 2012].

Physical acquisition methods interact directly with the device hardware, being able to retrieve unallocated (deleted) data, at the cost of using more invasive procedures. Additionally, there is a high probability of a target device being rendered useless after their execution. Among physical acquisition techniques, the Hex Dumping and Joint Test Action Group (JTAG) methods provide investigators with an easier way to access the raw information stored in the flash memory. Hex Dumping is conducted with the use of special devices, known as flasher boxes, which are responsible for creating a hexadecimal RAM copy [Luttenberger and Creutzburg, 2011]. The JTAG method derives from the standard which bears the same name (Joint Action Test Group), a universal, manufacturer-independent interface with semiconductor chip support. This particular method requires the attachment of a cable or a wiring harness to a JTAG header or connector on the mobile device, being significantly more invasive than Hex Dumping. There are plenty of commercial and open-source forensic tools with physical acquisition features, such as Cellebrite Universal Forensic Extraction Device (UFED) [Cellebrite, 2018], EnCase Forensics [Guidance Software, 2018], NowSecure (formerly ViaExtract) [NowSecure, 2016] and CDMA Workshop [CDMA Software, 2018].

Also considered as a physical acquisition method, chip-off techniques involve direct data retrieval from non-volatile memory chips of the target device. Data are extracted as an adjoining file in binary format, by reverse-engineering the wear-leveling flash algorithms. This method is also considered invasive, incurring in a higher risk of causing irreversible damage to the device. Some of the forensic tools supporting chip-off are Soft-Center NAND Flash Reader [Soft Center, 2018], BeeProg2 [ELNEC, 2018] and NFI Memory Toolkit [Netherlands Forensic Institute, 2018].

Micro Read is the latest addition to the existing methods [Murphy, 2013]. It involves the use of an electron microscope in order to observe the gates on a NAND or a NOR flash memory chip. Its usage is not publicly disclosed and it is currently limited to the extreme cases of national and international security crisis.

Logical acquisition is performed by establishing a connection between the device and a forensic workstation via a wired or wireless link; the appropriate security precautions are also taken. Such methods interact with the mobile device file system [Casey, 2011] to extract bitwise copies or memory segments. Contrary to physical acquisition methods, they are incapable of retrieving deleted files, being less invasive. Many forensic tools with physical acquisition features also support logical acquisition (Cellebrite UFED [Cellebrite, 2018], NowSecure ViaExtract [NowSecure, 2016]), while others have solely logical acquisition features, such as Autopsy [Autopsy, 2016] and Nyuki Forensic Investigator [Silensec, 2016]. Pseudo-physical acquisition is performed with the use of a bootloader, which alters only the protected area of the device (e.g. RAM) where it is uploaded [Klaver, 2010]. File system access is performed either by a logical dump on the phone's memory partitions [Hoog, 2011] or by access to the OS's databases.

Live acquisition deals with near real-time content extraction. It allows dumping parts of the runtime mobile device execution environment, such as the kernel process list, the kernel hash table [Hanaysha et al., 2014] and logs, so as to acquire evidence that would otherwise be lost after a potential device shut-down. It is divided into the network-based and volatile memory subcategories.

Live acquisition procedures take place between the two prevailing non-persistent elements of the mobile device, i.e. the volatile memory and network data. For the first case, the most common approach employs a modified bootable kernel [Volatile Systems, 2011], albeit a less invasive technique is also used by the Nyuki Android Process Dumper (AProcDump) [Silensec, 2016]. This particular implementation consists of an executable running on Advanced RISC Machine (ARM) Android devices [Nguli et al., 2014], which performs a dump of all running applications. The tuples of the dumped applications and their process ids are then saved in a file for future association to events and other activities of forensic interest. For the network data case, live forensics can also be applied and acquisition takes place either by direct access to the network interface and the packet buffers, or indirectly, via an application. Linux Memory Extractor (LiME) [504ensics Labs, 2013] is claimed to have this particular functionality [Heriyanto, 2013].

The non-intrusive forensics category encompasses the simplest forms of retrieval, classified between the observation and interaction categories. Observation techniques include "whatever an individual is capable of acquiring from a device via direct interaction with the installed applications" [Mokhonoana and Olivier, 2007] and their manual registration or via third party recording [Grispos et al., 2011] with a digital camera. This approach has three major drawbacks: it can become extremely time-consuming when the amount of data to be extracted is relatively big; it is totally ineffective when the device screen is destroyed; and finally, the acquisition accuracy is neglected, due to the probability of human error. The interaction category makes use of physical or biological traces on a device, such as fingerprints and DNA or other damage types that may be used as evidence upon court.

Due to several factors, the forensic acquisition method landscape is constantly expanding towards new directions and changing over time. The increased popularity of live forensic techniques is such an example, as they provide a way to overcome the limitations of post-mortem forensics, enabling the acquisition of volatile elements. Live forensic techniques are also significant for the conduction of investigations in the Cloud environment, which will be thoroughly examined in the next subsection.

2.2.2 Cloud Computing and Mobile Forensics

As a result of the increasing need for flexible computing power and storage capabilities while reducing infrastructure costs, organizations are migrating to remote, virtualized and on-demand services [Grispos et al., 2012], known as Cloud Services. They offer "virtually unlimited dynamic resources for computation, storage and service provision" [Khan et al., 2014].

The Cloud Computing paradigm is defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management efforts or service provider interaction" [Mell and Grance, 2011]. Cloud Services provide the means for organizations to scale their IT infrastructure with a level of efficiency, agility and flexibility which is difficult to meet solely with in-house resources.

Currently, the prevailing models related to Cloud Services are: "*Software-as-a-Service (SaaS)*, *Platform-as-a-Service (PaaS)* and *Infrastructure-as-a-Service (IaaS)*" [Ninawe and Ardhapurkar, 2014], often referred together as the Cloud Stack. SaaS applies to the cases where *Cloud Service Providers (CSPs)* offer applications to the clients, often accessible via a web browser, thus dispensing the need for software distribution or deployment. In the PaaS model, users' flexibility and control levels are relatively higher, since they are able to create and distribute their applications using an *Application Programming Interface (API)* and even manage their own databases. Lastly, in the IaaS model, clients can lease virtualized servers, where they can setup whichever type of virtual machine suits their needs. They also may have partial control over network infrastructure, such as firewalls and other solutions. There also exist other service models, such as *Database-as-a-Service (DBaaS)*, where users "store their data in a key-value pair" [Motahari-Nezhad et al., 2009] or even *STorage-as-a-Service (STaaS)*, which is exclusively dedicated to users' data handling, allowing them to store, download and share their data [Shariati et al., 2015]. Some of the most popular STaaS solutions are Dropbox, Box, Microsoft OneDrive, Google Drive, SugarSync and Ubuntu One.

From a cyber-security perspective, Cloud Computing also has its own downsides. Cloud Services can be used to support criminal activity, by spreading different malware types, or even by providing Crimeware-as-a-Service. Moreover, the multi-tenancy capabilities used to support concurrent virtual infrastructure management also contribute for hampering tracing procedures, and subsequently, promoting Cloud-based crime [Zawoad and Hasan, 2013].

Despite being the future of Internet services, Cloud Computing is also the future of electronic crime.

Beyond desktop or notebook computers, Cloud Services are increasingly involved in providing infrastructure, resource and complementary service needs for the mobile device consumption. An estimation for the next two years (2017-2019), predicts that the average market share of Cloud applications worldwide will reach 13,7% [Pang, 2015]. This poses a challenge for MF investigators, who have to account for the usage of Cloud Services in the investigation process.

The adoption of Cloud Services has led to the creation of a specific forensic discipline, defined as “the application of digital forensic science in Cloud Computing environments” [Badger et al., 2012]. This discipline can be defined from several different perspectives. “Technically, it consists of a hybrid forensic approach geared towards the generation of digital evidence” [Samet et al., 2014]. From an organizational aspect, “it involves interactions among Cloud actors for the purpose of facilitating both internal and external investigations” [Farina et al., 2015]. From a legal standpoint, it often involves “dealing with multi-jurisdictional and multi-tenant situations” [Ruan et al., 2013].

Cloud investigation involves forensic operations both in the Cloud and the equipment sides, requiring the use of CF, MF and NF techniques. Even though investigation in mobile devices can be accomplished by applying already existing forensic methods or tools, the same cannot be said about Cloud resources. Most post-mortem forensic tools have limited capabilities over Cloud-hosted data. This constitutes a challenge for the DF discipline, since users are increasingly relying on Cloud Services, decreasing the amount of forensically relevant data hosted on mobile devices. This situation is leading researchers towards alternative approaches, based on the use of live acquisition and interaction techniques. However, these features are not yet referenced by the existing standards [Ayers et al., 2014], as Cloud Forensics is a developing discipline, yet in its early stages.

The impossibility of gaining physical access to the Cloud infrastructure constitutes an impediment to the investigators’ work [Marturana et al., 2012], aggravated by the fact that Cloud data are frequently spread among various locations on different countries – often with different legal jurisdictions. The high volatility of virtual infrastructure logs creates an additional problem, as this information is vital for non-repudiation purposes. Finally, another dilemma in the field of mobile Cloud Forensics relates to the need to ensure network device connectivity during an investigation process, without risking a remote wipe or data alteration from a potentially compromised CSP.

Overall, the Cloud Forensics discipline requires new procedures to be developed for evidence acquisition, while avoiding data loss or corruption. For instance, Cheng [Cheng, 2011], proposes a Cloud-based engine, responsible for monitoring information flows and network traffic via interaction with various Cloud nodes. The mechanism aims to collect evidence from volatile (data related to the virtual infrastructure of the CSPs) or non-volatile data. Additionally, Chung et al.

[Chung et al., 2012] describe an investigation procedure for cases involving the use of Cloud storage services. It begins with the acquisition of an OS image from the target device, according to platform-specific procedures, which is later parsed for Cloud-storage application-related artifacts. If such artifacts are present, legal procedures (such as requesting search and seizure warrants or international judicial assistance) are taken, in order to proceed with further data analysis and reporting.

However, the Cloud is not only a source of challenges for digital investigation; it can also provide several benefits. When designing forensic solutions for mobile devices, aspects concerning computational and energy trade-offs have to be taken into consideration, because they are responsible for limiting the incorporation of specific functionality. Cloud technologies can be used to support and improve the efficiency of forensic tools, providing the required computing resources (such as data processing or storage) in a flexible and on-demand fashion [Lee and Hong, 2011]. Furthermore, “current forensic investigation requires correlative analysis of multiple devices and previous cases” [Lee and Hong, 2011]. This procedure is rather time- and resource- consuming and can be streamlined by taking advantage of Cloud Computing resources.

Also, Cloud-based forensic tools could eventually help to alleviate the problem of heterogeneous mobile OS platforms. Such platforms are substantially different among them, requiring different approaches for developing forensic tools. For this reason, most developers prefer to target popular device ecosystems (such as Android and iOS), for whom a plethora of tools exist. A Cloud-based tool could provide a potential platform-agnostic solution to this particular issue, enabling data acquisition and analysis even from devices that belong to less representative platforms (i.e., with a smaller market share).

Despite the fact that Cloud Computing is becoming a mature and widely used discipline, Cloud Security and Forensics need substantial improvement. In the next subsection, surveys related to CF, NF, MF and their equivalent Cloud Computing contributions are presented.

2.2.3 Mobile Forensic Practices

The current section provides a literature review of publications about forensic practices that are directly related or relevant to the MF field. The key for developing effective MF tools and methods is a deep and detailed understanding of the field, with a particular focus on two factors. First, technical knowledge, acquired either theoretically through research, or by actual practical involvement with the subject. Second, the use of logical or mathematical languages, aiming to model the field’s basic elements. Despite the specific characteristics of each domain, there are several different approaches covered in surveys on CF and NF techniques that can be transposed to the MF scope, thus remaining relevant to the latter context.

One of the most exhaustive surveys on forensic techniques was presented by Kohn et al. [Kohn et al., 2013]. The authors gathered and formalized into pseudo-code the existing scattered process models for digital forensic investigation and provided their comparative summary. Moreover, they introduced a process model of their own, called *Integrated Digital Forensic Process Model (IDFPM)*, which addresses some of the more persistent investigation issues by schematically organizing the most critical steps in a timeline. IDFPM and forensic process models in general can serve as a base for new formal models involving additional concepts, such as Cloud Computing.

In the field of NF, the survey by Pilli et al. [Pilli et al., 2010] provides a complete view of the evolution of this discipline, the existing *Network Forensic Analysis Tools (NFATs)* and related research challenges. Moreover, the authors introduce a novel generic process model for NF.

Specifically for MF, Barmpatsalou et al. [Barmpatsalou et al., 2013] provide a state-of-the-art study on forensic techniques, updated for smartphone-era devices. Besides describing MF standardization efforts, they also classify existing research, which is presented in a timeline according to the acquisition type, mobile OS, low level modifications (root-jailbreak) and acquired data types. The aim of such a representation is to aid future researchers to locate research trends within a time context and to observe the evolution of MF through time.

Martini and Choo [Martini and Choo, 2014] conducted a literature survey on cases involving Cloud Services as sources of evidence. The survey examines technical or conceptual Cloud-aware solutions for collection of forensic evidence. It also includes works related to analysis of specific Cloud-based products and services (Dropbox, OneDrive). The authors analyze the data types that can be acquired directly from a Cloud Service and focus on what can be retrieved from a device after interacting with Cloud applications.

An overview of the current research trends in the intersection of the fields of MF and the Mobile Cloud was provided by Samet et al. [Samet et al., 2014]. The authors enumerated the most significant mobile Cloud Forensics challenges, such as limitations of post-mortem and live forensic tools or limited investigator control over the device and legal issues. Since mobile Cloud Forensics is a relatively new discipline without much dedicated research, they included references related to computer Cloud Forensics, with a potential application to the mobile domain.

A survey on the trends and future challenges of MF concerning the fourth quarter of 2014 was published by Cellebrite Predictions [Cellebrite Predictions, 2015]. Despite not being a purely academic work, it provides useful metrics concerning the state of forensic investigations. Among others, it is mentioned that the most significant data sources are (by descending order of relevance): the mobile devices themselves, third party applications, wireless, cellular providers and CSPs. Moreover, device and application encryption, data stored in CSPs and big data manipulation are considered as the most prominent emerging challenges.

In their survey, Ardagna et al. [Ardagna et al., 2015] expand the concept of Cloud Security towards Cloud Assurance. They claim that assurance as a notion is the expectation that security measures taken will be as effective as initially planned. While security consists of the implemented solutions for system protection and threat prevention, assurance incorporates techniques concerning evidence collection and analysis. Moreover, they present various Cloud Security solutions and their Cloud Assurance equivalents.

Kechadi et al. [Kechadi et al., 2015] conducted a survey on forensic investigations in the Cloud Computing ecosystem. Initially, the authors identify the resource and computational trade-offs in contemporary mobile devices and highlight the significance of the mobile cloud computing discipline. Additionally, they enumerate the potential challenges that may arise during a forensic investigation in the Cloud environment. They also present the differences between traditional and cloud-based mobile forensic techniques within the investigation process. The paper concludes with a presentation of some state-of-the-art milestones about application of forensic methodologies in Cloud storage services with mobile device involvement.

The current part of the background work analysis observes how the MF discipline evolves over time. It covers a wide span of areas of expertise that are considered MF sub-disciplines and includes them in a special taxonomy scheme which integrates older and contemporary research papers in a flexible manner. A more detailed overview is presented in the following section.

2.3 Recent Advances in Mobile Forensics

The field of research in MF has shown an admirable amount of growth over the last seven years. Ntantogian et al. [Ntantogian et al., 2014] already proposed a preliminary classification regarding research directions. Also in this line, Kaart and Laraghy [Kaart and Laraghy, 2014] provide additional insights about expanding the research of MF further from acquisition methodologies. Beyond data acquisition or analysis, other emergent MF concepts and methodologies are beginning to appear in related literature. The subject of these papers falls along five main categories, namely:

1. File acquisition and data integrity
2. Identification of malicious activity and malware analysis
3. Evidence reconstruction and presentation
4. Evidence parsing
5. Automated classification and analysis of user and application behaviour

File acquisition has been one of the very first concerns among MF researchers, since the acquisition phase is a critical part of the investigation process model, constituting the initial information gathering procedure. During this phase, investigators also have to maintain data integrity so as to preserve evidence admissibility upon court. No further actions can be taken during an investigation if acquisition is not properly performed and retrieved content is not validated.

Beyond forensic purposes, evidence retrieved from mobile devices can also be useful for cyber-security analysis. When target devices are attacked, compromised by malware or forced into becoming part of a botnet, data acquired from them can provide useful insights to security professionals concerning behavioral patterns and signatures of malicious software. Post-mortem device analysis or live examination can be performed so as to achieve identification of malicious activity and further malware analysis [Casey, 2013].

Evidence reconstruction and presentation is another rising concern in the MF research world, since evidence presentation modeling aids the investigation procedure. Interestingly, Kasiaras et al. [Kasiaras et al., 2014] noted the existence of an unbalanced distribution between the amount of research papers corresponding to data acquisition and integrity preservation and the number of papers concerning presentation of evidence and further facilitation of the investigators' role.

Evidence parsing is mainly related to the parsing and decoding of acquired data. Due to the wealth of available tools and resources for this purpose, this area has been lagging behind in terms of available research. Moreover, even though current solutions are not exactly suitable for every purpose, it is not difficult for an individual to create a customized script for file parsing.

Despite the fact that the need for such methods has been highlighted relatively early [Marturana et al., 2011], few research papers have been published towards that direction. However, the use of automated procedures, based on technologies such as Machine Learning algorithms, Fuzzy Systems and Neural Networks would not only facilitate investigations, but also automate many procedures without the need for continuous experts' supervision. The following subsections elaborate the advances in each one of the aforementioned categories.

2.3.1 File Acquisition and Data Integrity

This subsection comprises two different families of techniques: conventional (or classic) techniques, that acquire information from standalone devices and Cloud-aware techniques, which are oriented towards the incorporation of Cloud Service awareness.

2.3.1.1 Conventional Techniques

Thing et al. [Thing et al., 2010] presented an automated mechanism for retrieving volatile memory parts from Android devices. The authors developed memgrab, a memory acquisition tool, which tracks process IDs and memory addresses "from the procfs virtual file system provided by the kernel" [Thing et al., 2010]. Once elements related to the processes are extracted, a Perl-based script, named memory dump analyzer, searches for the needed evidence elements.

Dezfouli et al. [Dezfouli et al., 2012] proposed an acquisition method of volatile memory contents in Android devices, which claims minimal data modification when compared to existing alternatives. A part of the non-volatile memory of the device is reserved for storing the information deriving from the process acquisition mechanism. The technique involves updating the initial dump by using the deltas (i.e., the different parts) from consecutive captures.

Aiming to extend the research horizon of iOS acquisition methods, Gomez-Miralles and Arnedo-Moreno [Gomez-Miralles and Arnedo-Moreno, 2012] proposed a technique for iPads based on the Camera Connection Kit. The authors claim that this method, equally to the one proposed by Zdziarski [Zdziarski, 2008], is less invasive in terms of device data alteration. They also highlight the need for data acquisition techniques that are more complete than their predecessors, such as the iTunes backup, which is not capable of retrieving unallocated data. Their pseudo-physical acquisition method consists of the following steps: jailbreaking the device in order to gain administrative rights; installing openssh (Secure Shell (SSH) Server) and coreutils libraries; deactivating the network auto-lock feature; connecting the device to a Hard Disk Drive (HDD) by the Camera Connection Kit; and finally performing a disk duplicate (dd) command. One of the advantages of the proposed method is that despite the existence of device encryption (in iOS 4), most of the acquired files can be decrypted since the key stored in the device is acquired as well. The authors conclude by evaluating the solution and expressing their concern about the next generation encryption layers and the private user encryption keys that could not be acquired.

Kotsopoulos and Stamatiou [Kotsopoulos and Stamatiou, 2012] discuss the problem of forensic data acquisition in the simultaneous presence of countermeasures. Their existence may become an impediment for the investigators, since data obfuscation, alteration and detection of forensic tools are able to hamper their work. The authors suggest a consolidation of open source tools for acquisition of volatile content and encryption key detection, that aims to reveal potentially malicious content hidden in encrypted files.

Data encryption and its effectiveness against potential eavesdroppers is discussed by Al Barghouthy and Said [Al Barghouthy and Said, 2013]. The authors performed logical forensic acquisition in an Android device after using Instant Messaging (IM) applications, private browser sessions or social media over the latter. They attempted to examine the actual readability of artifacts from messaging applications with and without applied encryption. While additional encryption is proved effective in the majority of social media message exchange, it can also hamper DF procedures for the same reasons.

Votipka et al. [Votipka et al., 2013] introduced a modified boot image for Android devices in order to balance between the potential data loss arising from logical acquisition methods and the invasive tactics of physical acquisition strategies. As a result, an alternative, device-agnostic version of an Android boot mode was proposed. More precisely, before proceeding to acquisition via recovery mode, the presented methodology incorporates a software collection package, which gathers the essential elements for booting the specific target device.

Cold-boot attacks in Android smartphones were introduced by Muller and Spreitzenbarth [Müller and Spreitzenbarth, 2013], when the authors proved that data in smartphone RAM chips fade in a slower pace once the device remains frozen for a certain period of time. They also introduced Forensic Recovery Of Scrambled Telephones (FROST) recovery image, a custom bootloader which was flashed on the device after the cold-boot attack and provided the potential investigators with the options of acquiring encryption keys, brute-force attacking weak user passwords and unlocking and accessing the user data partition.

Most of the classic acquisition techniques introduced during the last seven years are experiments in novel fields, an encouraging fact for the future of MF. In the following subsection, more details about techniques which are destined for a cloud environment can be encountered.

2.3.1.2 Cloud-aware Techniques

Apart from describing the current trends concerning DF in the presence of Cloud Services, Marturana et al. [Marturana et al., 2012] created a case study of forensic acquisition from Cloud Service Providers (CSPs) in a Windows 7 environment, with the use of already existing methods. The described use case consisted of a forensic acquisition procedure performed at the client side after interacting with various CSPs. Despite the fact that the research is not purely related to mobile devices, the described methodology has potential for future use in such an environment.

One of the main risks present in the Cloud is associated to information volatility. If information is not acquired within a specific time window, its integrity can be compromised, since it is often impossible to be aware of which entities have accessed, altered or deleted the cloud data by the time the investigation began. As a solution to this issue, Zawoad et al. [Zawoad et al., 2013] introduced a method that stores "virtual machines' logs and provides access to forensic investigators, ensuring the cloud users' confidentiality, named Secure Logging as a Service" [Zawoad et al., 2013].

The probability of mobile devices serving as proxies for data leaks from Cloud Services was addressed by Grispos et al. [Grispos et al., 2013]. Dropbox, Box and SugarSync Cloud storage services were used as testbeds. After conducting physical acquisition on various Android and iOS devices, with different usage scenarios, the obtained images were examined for artifacts related to the Cloud Services.

One of the approaches to cope with Cloud Services in a forensic context favors the introduction of continuous monitoring techniques to gather information for DF purposes. In this line of reasoning, Grover [Grover, 2013] implemented Droidwatch, a prototype monitoring system for Android devices. Droidwatch consists of an on-phone application and a remote enterprise server. The application tracks the occurrence of incidents in the device and reports them back to the server. The system is also equipped with a mobile database, which gets unloaded upon information syncing with the remote instance. The collection of datasets makes the tool an interesting aspect for security auditing, forensic investigations and MDM, especially in BYOD scenarios.

Baggili et al. [Baggili et al., 2015] performed a forensic retrieval procedure in two smartwatches. The authors used a variety of proprietary and open source forensic tools, popular in the MF community. Their preliminary research showed that information of critical forensic interest that does not usually reside in mobile handsets, such as data from heart rate monitors and pedometers, can be acquired through a relatively easy process. Wearable devices upload data concerning the users to the Cloud for monitoring and processing purposes and their acquisition provides the investigators with potentially high quality evidence.

Daryabar et al. [Daryabar et al., 2015] experimented with the MEGA Cloud storage mobile client application. Their research key points included detection of alterations in files and metadata used by the application and discovery of forensic evidence in target devices running iOS and Android. The authors used an adapted version of the investigation framework proposed by Martini and Choo [Martini and Choo, 2012]. Afterwards, they retrieved a bitwise copy of the Android Jellybean 4.2 internal memory and extracted an iTunes backup from the iOS 7.1.2 handset. TCPDump and Wireshark were used for capturing sent and received network packets. Additionally, they created an experimental scenario which included the following interactions with the MEGA mobile application: logging in with a set of custom credentials, uploading, downloading and deleting different files and sharing a file to a custom e-mail address. MD5 hashes and timestamp comparisons were applied in order to detect changes occurring to the uploaded and downloaded files. While “MD5 values of the original and downloaded files matched” [Daryabar et al., 2015], timestamps showed a certain level of instability and they were always dependent to the date and time settings of the target devices. Installation activity and usernames from the logging in activities were encountered in both devices. Moreover, the authors were able to detect a non-encrypted file version of the password used in the Android device. Data corresponding to upload activity was only tracked in the iOS device, whereas download activity data was present in both devices. The name of the deleted file was present in both devices as well. Lastly, no elements concerning the sharing activity were present in any device.

Adversary models are a known practice in the field of Information Security and Cryptography, with little or no expression in terms of DF- or MF-related research activity. In a novel approach, Do et al. [Do et al., 2015] created an adversary model aiming to collect and analyze data from six widely used Cloud applications. With respect to the principles of forensic soundness, such as keeping any device modification to a bare minimum to avoid interference with the forensic process, the authors developed an acquisition and analysis methodology based on this model. Its main innovative characteristic results from combined factors: using a live OS; avoiding modification of the boot or recovery partitions to securely acquire evidence; and providing data analysis capabilities.

A complex study on evidence acquisition of Cloud storage applications in mobile devices was performed by Grispos et al. [Grispos et al., 2015]. The authors used practitioner accepted forensic tools, such as the Cellebrite UFED or the Forensic Toolkit (FTK) Imager and FTK Toolkit, in devices running iOS and Android. Their concerns went beyond the data recovery process and how it can be affected by the usage of Cloud Services. They also encompassed aspects such as: how different application versions alter the acquisition outcome; what acquired metadata reveal about remote storage at the provider's side; and if the evidence retrieved from two different versions of Cloud application sources provides a more detailed dataset of results. One of their most useful findings is that data acquired from a mobile device can serve as a snapshot of the CSP-hosted data.

Even though changes such as file deletions may occur in the future, an acquisition prior to that precise moment constitutes proof that the file existed beforehand. Moreover, the way in which the device is used is able to affect the acquisition outcome. For example, fewer files are recovered if the user had previously performed a cache cleaning. It was discovered that different Cloud storage application versions lead to a variation in the number of acquired files. Additionally, information stored in metadata could be used to hint at the storage state in the CSP side, or even to give access to download files that did not exist in the acquisition data. The methods adopted by this study could be easily used with contemporary OS versions, different Cloud storage application versions, as well as with applications simultaneously hosting and monitoring multiple Cloud storage accounts.

Martini et al. proposed "a device-agnostic evidence collection and analysis methodology for mobile devices" [Martini et al., 2015a]. The authors use a custom bootloader and a live OS to perform a physical dump of the device partitions – a sound approach from a DF perspective, also adopted by others. Afterwards, all the available applications are obtained and different locations are explored for data of forensic interest. Practical use cases involved acquisition and analysis of seven popular Android applications for Cloud-storage, password sync and notes (Dropbox, OneDrive, Box, ownCloud, Universal Password Manager, EverNote and OneNote [Martini et al., 2015b]) to retrieve evidence of forensic interest (including sensitive data such as credentials and authentication tokens) in private and public application storage locations.

Shariati et al. [Shariati et al., 2015] investigated the effectiveness of forensic acquisition for artifacts of the Ubuntu One Cloud storage service in devices running Windows 8.1, MacOSX 10.9 and iOS 7.0.4. The explored use cases covered the acquisition of artifacts after accessing a Cloud Service via its own application and by browser access. Volatile content and network artifacts were examined separately. While traces of application usage were present in every platform, the same cannot be claimed for sensitive data, such as credentials and authentication tokens, whose vestiges varied among different platforms. Recently, Shariati et al. [Shariati et al., 2016] conducted a similar study concerning the SugarSync service and included Android in the list of the test platforms, obtaining similar results.

A comparative study concerning the acquisition and discovery of forensic artifacts between Android and Windows Phone devices was conducted by Cahyani et al. The authors distinguish three different use cases of "Cloud storage and communication applications, namely information propagation, information concealment and communications" [Cahyani et al., 2016b]. The first case is related to signing in, accessing and downloading files saved in Cloud storage services. The second case corresponds to exchange of files modified by a steganography technique via e-mail, communication (Skype, WhatsApp, Viber) and Cloud storage applications. Lastly, communication applications are used as means of information exchange and activity (friend addition, chat) tracking. The authors used physical, logical and manual acquisition techniques, depending on each method's applicability on the different target devices. Logical acquisition of Android devices resulted in successful retrieval of user credentials, actions and downloaded data. On the contrary, only the latter pieces of evidence were available in devices running the Windows Phone operating system. The authors concluded that only physical acquisition is capable of retrieving a significant amount of artifacts from Windows Phone smartphones, thus cross-validating the assumption made in one of their previous papers [Cahyani et al., 2016a].

In the last few years, an explosion in the use (and misuse) of mobile services, -with and without Cloud contribution- was observed. One of the fields strongly correlated to criminal activities that is also in need of new, adaptable mechanisms and continuous surveillance is the one of malicious activity identification, which is presented in the next subsection.

2.3.2 Identification of Malicious Activity and Malware Analysis

This subsection analyzes live methods, which occur in real-time, and classic methods, which take place upon malware infection.

2.3.2.1 Live Methods

Taking into account the energy and processing trade-offs that occur when a continuous monitoring application is running, Houmansadr et al. [Houmansadr et al., 2011] introduced a high level architecture of a Cloud-based Intrusion Detection System (IDS). This IDS uses a Cloud proxy server in order to perform off-loaded forensic analysis and malware recognition on the extracted data from the device. However, the practical feasibility of such a method is debatable and requires further research and experimentation, mainly due to the large amount of exchanged data.

The Volatility Framework is a multi-platform memory forensics solution. Whether the extracted memory product is in “raw format, a Microsoft crash dump, a hibernation file, or a virtual machine snapshot” [Volatile Systems, 2011], it provides the investigators with a complete view of the examined system. Identification of malicious activity was one of its initial concerns, but nowadays its functionality has expanded. Since the release of version 2.3.1 in October 2013, support for Android kernels was also added, expanding the potential of Android forensics to a new level.

A fake, intentionally set up GSM/GPRS network was created by Schutz et al. for intercepting network traffic to and from a potentially compromised mobile device [Schutz et al., 2013]. This way, network traces get trapped and are further processed by the Wiretrap application.

Frequently, criminal actions are directly associated to device compromising from malware or third party attacks. Analysis of audit data from forensic associations can help investigators to create behavioral patterns of several mobile device threats. This can be achieved by exploring “hidden processes, their structure, suspicious executed code and other entities” [Hanaysha et al., 2014]. The creation of an open source Android memory forensic investigation environment was the main subject of the solution proposed by Hanaysha et al. [Hanaysha et al., 2014]. Focusing on live acquisition, the authors used a combination of the Volatility Framework with LiME, aiming to identify and trace the assets compromised by malware. They simulated use cases by installing common Android malware, such as the O Bada Trojan and ZitMo in the target device. By accessing hidden processes and gathering information about their structure from the kernel process list, the kernel hash table and the kmem_cache, they were able to trace them back to the malware activity. However, an automated version of this procedure is yet to be researched.

Borges et al. [Borges et al., 2017] introduced HyIDS, an Android hybrid IDS, that aims to provide a solution for the BYOD principle in PPDR scenarios. HyIDS consists of a host and a remote module. The host module is installed on a target device, where it performs data collection from the Android logging services and securely forwards this information to the remote module, the Command and Control Centre (CCC). The CCC performs static and dynamic malware analysis and uses a correlation engine so as to trace events that do not comply with the concerned PPDR organization’s policy.

Even though live methods are indisputably the future of the race against malware, classic methods – presented in the next subsection – can still produce meaningful contributions.

2.3.2.2 Classic Methods

Di Cerbo et al. [Di Cerbo et al., 2011] presented the functionality of a forensic tool (AppAware), especially designed for detecting Android malware based on permission abuse. As soon as the developed application is executed on the device side, it generates an eXtensible Markup Language (XML) file containing the permissions requested for the selected application. Then, the investigators are prompted to manually compare the results to a classified list of potential malware. Despite the fact that the particular forensic tool is relatively useful, automated comparison and classification of the malware would be a considerable evolution.

A typical guideline for recognizing mobile malware via a forensic procedure consists of the following steps: identification of suspicious programs, neutralization of any anti-forensics code, code extraction from the malware body, and deduction of malicious functions [Li et al., 2012]. In this paper, the authors propose a method of identifying mobile malware via reverse engineering, analysis and reconstruction of events related to their functionality.

Eradicating malicious activity at the highest possible scale can be rather characterized as an achievement. Nevertheless, mobile criminology is not solely dedicated to the malware identification and taken countermeasures, but to the potential crime scene as a whole. The next subsection introduces the subdiscipline of evidence reconstruction and presentation as a means of facilitating the investigators' duty.

2.3.3 Evidence Reconstruction and Presentation

This subsection enumerates and analyzes research papers concerning the reconstruction of evidence deriving from forensic data. It presents two different groups of approaches: event presentation as a whole and reconstruction of specific elements.

2.3.3.1 Event Presentation

While aiming to enrich the chronological evidence presentation for forensic tools, Kasiaras et al. [Kasiaras et al., 2014] created the Android Forensic Data Analyzer (AFDA). Its operation is summarized in two phases. During the first phase, the tool executes common forensic investigation tasks, such as image mounting, evidence retrieval, hash creation and report generation. The second phase consists of a timeline where the events associated to the acquired evidence and their correlated assets are presented in a chronological order. Moreover, it provides the investigator with the option of tracing the exact location history of the device by parsing geodata used by many different applications.

Zawoad and Hasan [Zawoad and Hasan, 2015] created a conceptual model of a mechanism responsible for preserving (in a secure database) and presenting (via GET requests to a "Representational State Transfer (REST) API" [Fielding, 2000]) data acquired from mobile Cloud investigations. The model was designed after enumerating the challenges the field of mobile Cloud Forensics is facing and highlighting the requirements for secure mobile Cloud transactions.

The presentation of events that occurred during the conduction of a crime is undoubtedly a useful element. Its effectiveness is complemented by the reconstruction of evidence and other elements, which is elaborated below.

2.3.3.2 Reconstruction of Specific Elements

IM applications contain significant data for the outcome of an investigation. Reconstructing the information from various points of an Android device memory image has been the primary concern mentioned by Anglano [Anglano, 2014]. Apart from the reassembly task, the author attempts to correlate various different events and timestamps related to the forensic artifacts.

Law enforcement agents, judges and prosecutors need to have detailed answers to the questions rising when a series of incidents takes place. Kaart and Laraghy [Kaart and Laraghy, 2014] highlight this importance and perform a case study on detecting the intentional clock skewing in an Android device, by accessing the mmssms.db database.

The paper by Saltaformaggio et al. [Saltaformaggio et al., 2015] introduces GUITAR, an application-independent method capable of reconstructing Android application Graphical User Interfaces (GUIs) from their memory heaps, which reside in a forensically acquired memory image. The method uses a "depth-first topology recovery algorithm" in order to sort the fragmented application hierarchy. Afterwards, the application graphical pieces are united in the correct order with the aid of a "bipartite graph weighted assignment solver and a drawing content-based fitness function" [Saltaformaggio et al., 2015]. In the end, a windowing system binary is used so as to create the final form of the redrawn application.

The next subsection discusses the recent advances in evidence parsing, one of the most fundamental concepts in the field of MF.

2.3.4 Evidence Parsing

The research works discussed in this subsection have two different objects of study. The first category contains data from social media and messaging applications, while the second category concerns various data and focuses on personal and hybrid information from various sources.

2.3.4.1 Messaging Data Parsing

Investigation for Skype artifacts in the NAND and RAM memory of mobile devices running the Android OS was performed by Al-Saleh and Forihat [Al-Saleh and Forihat, 2013]. Live process dumping and logical acquisition methods were used and experiments took place with different Skype usage scenarios. Evidence parsing for Skype usage traces was performed manually and by utilities such as the grep tool and the Eclipse Memory Analyzer. The research pointed out that elements concerning Skype activities remain in both memory types and can be traced even after deletion.

In a mobile device forensic investigation, all acquired evidence should be taken into consideration, handled and combined so as to reach a satisfactory conclusion. Data deriving from Social Networks (SNs) and their equivalent messaging applications are evidence sources that can facilitate an investigation process. Dezfouli et al. [Dezfouli et al., 2015] investigated SN applications in Android (4.2) and iOS (7.1.2) devices for elements of forensic interest. Examination of Facebook, Twitter, LinkedIn and Google+ applications revealed that the majority of the user-related data (such as usernames, profile pictures, posts and messages) other than passwords could be retrieved.

The next section describes the research conducted in a more diverse data type environment.

2.3.4.2 Personal/Hybrid Data Parsing

Data deriving from anonymizing services had also been a concern in the MF community. A case of forensic investigation of Orweb browser data in rooted and non-rooted Android devices was examined by Al Barghouthy et al. [Al Barghouthy et al., 2013]. Acquisition was performed by the general purpose Titanium Backup application instead of a dedicated forensic tool. The use of the latter might have different effects on the amount of collected information. Moreover, the use of a backup tool created an unnecessary impediment, since the backup utility only functions properly in rooted devices. As a result, an image of the non-rooted device could not be retrieved, a fact that could have been avoided if a forensic tool was used. On the other end, data acquisition from a rooted device was successful: databases were parsed with the SQLite Database Browser and artifacts such as URLs, Facebook IDs and chat conversations were identified.

The FROST recovery image mentioned in Subsection 2.3.1 was further improved by Hilgers et al. [Hilgers et al., 2014], by including the Volatility Framework [Volatile Systems, 2011] and the LiME plug-in [504ensics Labs, 2013]. With this addition, the authors were still able to access the device RAM in case the user data partition was wiped due to manufacturer security measures. They also managed to successfully parse the RAM for call logs, information typed by the user in a short timeframe before the cold-boot attack, Personal Identification Numbers (PINs), passwords and photo metadata.

Ntantogian et al. [Ntantogian et al., 2014] conducted experiments concerning the discovery of user credentials in different usage scenarios of various applications and use cases. The examination took place after live memory dumping of the target devices. The authors verified that as long as a mobile device remains powered-on, it is highly likely that some user credentials will remain in its memory. Findings from the specific research also unveiled the incapability of task-killers and password protection applications to safeguard sensitive personal information (or to wipe it, if necessary). The research also revealed many application vulnerabilities and simultaneously opened new future perspectives in data protection and prevention of anti-forensic techniques.

Immanuel et al. [Immanuel et al., 2015] highlighted the importance of searching various sources of information within a smartphone acquired memory image that may contain data of forensic interest. They created an Android caches taxonomy out of eleven installed applications of different kinds and modeled the classification process. Each cache type (WebView, SQLite, Volley, Serialized Java Objects, Network File and Custom) has different parsing methods. The authors developed a unified cache viewer application so as to facilitate the investigation procedure.

Evidence parsing is a structural element for automating various procedures related to forensic data analysis. However, a fully or partially automated solution requires more than simple parsing. The next subsection covers the advances in topics related to the automation of investigation processes.

2.3.5 Automated Classification and Analysis of User and Application Behaviour

The current subsection addresses research in the field of automation for digital investigation. It is organized in two categories: research works dedicated specifically to the discipline of MF, and general-purpose forensic methodologies, which can be applied to the MF field with some modifications.

2.3.5.1 Methodologies Related to MF

In an effort to optimize the investigation process during triage, Walls et al. [Walls et al., 2011] proposed DEC0DE, a library of Probabilistic Finite State Machines (PFSMs) based on successfully imaged devices. The tool operation includes two steps. First, the byte stream of an acquired physical image is inserted into a filtering mechanism of hashes belonging to previously examined devices in order to exclude a load of insignificant information. Second, the remaining data enter a multi-step inference component, based on a set of PFSMs so as to conclude the automatic recognition of critical data sequences, such as phone numbers, names, messages, photographs, videos, documents and audio clips.

Marturana et al. [Marturana et al., 2011] introduced a triaging method based on self-knowledge algorithms to predict user behavior and to classify mobile devices between suspects of content abusing or not. Their experiment consisted of tests with 21 different Android devices, applying three different Machine Learning techniques (Bayesian Networks, Decision Trees, Locally Weighted Learning) and validating the method that was initially used.

During their operation, mobile applications produce volatile and non-volatile traces that can be associated to the users' activities. Michalas and Murray [Michalas and Murray, 2016] introduced MemTri, a memory forensics tool based on the principles of the Volatility Framework [Volatile Systems, 2011]. MemTri uses regular expressions in order to identify illegal activity patterns from a seized memory image. Afterwards, a Bayesian Network is used to calculate the device owner's probability of criminal involvement. The specific tool is still being tested and the results of the experimental procedure were disseminated in late 2017.

2.3.5.2 General Purpose Forensic Methodologies

This section discusses some research papers with general purpose forensic methodologies which, nonetheless, may be relevant to MF.

Text mining and content clustering from documents were addressed by Nassif and Hruschka [Nassif and Hruschka, 2011]. The authors applied six clustering algorithms on text documents acquired from actual CF investigations and discovered verbal patterns that could aid future examinations conducted by experts.

Upon adoption of the Cloud as a forensic platform, Lee and Hong [Lee and Hong, 2011] propose a service named Forensic Cloud, which applies a logic-centered approach to the way a digital investigation is conducted – replacing the prevailing technology-centered approaches. The authors applied a model of index search on forensically acquired data, supported by a distributed system on cloud servers. Even though the indexing process is rather slow, the final result compensates the time spent. Moreover, their framework uses data abstraction techniques in order to provide a more realistic data representation to the investigator on the client side. Instead of bulk evidence listing, relevant data are grouped, thus facilitating decision-making and association procedures.

Platzer et al. [Platzer et al., 2014] introduced Skin-Sheriff, a method which uses Machine Learning techniques for detecting nude skin among acquired data. Despite not being dedicated to MF, this is applicable to every sub-discipline of DF where photographs are retrieved as evidence.

After the conclusion of the background analysis, the claim that automation in MF is a field in need of more contributions, is verified. Meanwhile, many authors are concerned with the lack of automated methods during the analysis phase of the investigative process model. Data analysis and classification are still performed mostly manually, leading to the need for further research towards the automation of such procedures. The investigation parts that are in need of automation have to be clarified and formalized. There are five main categories for which automation and application of hard and soft computing methods would be feasible:

- Data and artifacts classification
- User behavioural patterns and their adaptation
- Application and system related process categorization for potential discovery of malicious activity
- Correlation between incidents after data analysis from different sources
- Creation of logical rules and criminal profiles deriving from data patterns and respective profiling and Behavioural Evidence Analysis (BEA) so as to pinpoint towards specific crime types

The next section of this chapter is exclusively related to the concept of Mobile Forensic Data Analysis (MFDA) for digital criminal profiling and investigation.

2.4 Mobile Forensic Data Analysis and Digital Criminal Profiling

Mobile Forensic Data Analysis (MFDA) is a broad term that corresponds to the examination of evidence deriving from mobile devices with the final aim of reaching appropriate conclusions for different scenarios. MFDA can be performed for a considerable number of reasons, varying from plain observation to identification of malicious activity. In this thesis, we use a combination of MFDA with various criminal profiling principles, so as to encounter relationships and actions that constitute a digital criminal profile and later identify them in a dataset with the aid of Fuzzy Systems, Neural Networks (NNs) and the Adaptive Neuro-Fuzzy Inference System (ANFIS). Before elaborating the present Methodology, it is essential to introduce the digital criminal profiling principles that lead to its formulation.

2.4.1 Digital Criminal Profiling Principles

Criminal profiling is one of the techniques law enforcement organizations worldwide employ in order to solve traditional crimes. More precisely, it is “a technique whereby the probable characteristics of a criminal offender or offenders are predicted based on the behaviors exhibited in the commission of a crime”

[Kocsis, 2006]. However, with the rise of digital crime, new challenges and requirements for the law enforcement organizations appear as well. The following paragraphs present the digital criminal profiling methodologies that served as a springboard for the methodology used in this thesis.

Behavioural Evidence Analysis

BEA is the process of "deducing the psychobehavioral portrait of the offender based on professional training and previous investigations" [Ferrari et al., 2008] by a group of experts. It comprises four discrete steps, namely "equivocal forensic analysis, victimology, crime scene characteristics, and offender characteristics" [Mutawa et al., 2016]. Equivocal forensic analysis comprises the careful expert review of the case under investigation and requires the most objective reasoning in order to prevent the deduction of mistaken conclusions. Victimology analyses the victim's characteristics in order to infer what led to their choice. The characteristics of the crime scene concern every piece of digital evidence that can facilitate extracting conclusions for the case and finally, the offenders' characteristics are all the psychosocial, behavioural and physical characteristics that lead to the construction of a profile and match the criminals to crimes. Despite the objectivity requirement, this approach tended to be highly subjective, so a more consistent solution was needed.

Inductive Analysis

Inductive analysis has a more solid scientific base. The methodology aims to use previously committed crimes and data associated to them, so as to create a profile, the patterns of which are a match to yet to be discovered potential offenders. In inductive analysis, reasoning flows from the general to the specific. Data encountered in the crime scene are combined with the already existing theory for the production of hypotheses. The latter ones are then analyzed according to past investigations and expert knowledge in the field. Assumptions lead to the validation or alteration of the hypotheses and the scheme continues evolving according to the newer additions that influence the decisions taken. Fig. 2.4 presents the inductive analysis procedure.

Deductive Analysis

Contrary to inductive anlysis, the logical flow in deductive analysis moves from the specific to the general. Evidence deriving from the case under investigation is analyzed and a profile specifically dedicated to the actor of the case is constructed [Turvey, 2011]. Deductive analysis is not particularly efficient as a culprit identification methodology, but it serves more as an eliminating mechanism for groups of potential suspects.

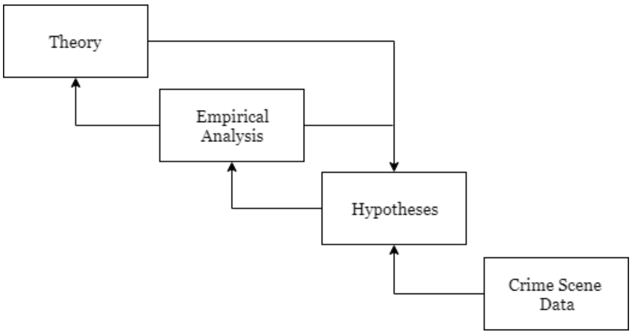

Figure 2.4: Inductive analysis, adapted from Godwin [Godwin, 2012]

CRISP-DM

CRoss-Industry Standard Process-Data Mining (CRISP-DM) includes the following steps: defining the problem and structuring a solution strategy; accumulating information from seized media and forming the datasets; performing feature selection; "matching of data in order to identify deficiencies, discrepancies or similarities" [Mena, 2003]; applying the knowledge retrieved from the data so as to construct a cybercriminal profile and evaluating the validity of the new profile.

The Intelligence Cycle

In 2011, the United Nations Office on Drugs and Crime (UNODC) published a guide on generic criminal intelligence analysis, applicable to a variety of cases. One of the basic models introduced in the publication was "The Intelligence Cycle", a circular scheme consisting of seven phases. The first phase, *Tasking*, is related to the crime under examination, the motivations that led to its conduction and the offenders' motivation. The *Collection* phase is a "formally defined approach to describing the information needed and the means of acquiring it" [UNODC, 2011], whereas the *Evaluation* phase is responsible for reassuring if the aforementioned information is reliable and in an appropriate state so as to constitute sufficient evidence. *Collation* is responsible for organizing and converting the information to an editable data format. The data are thoroughly examined and important features are highlighted during the *Analysis* phase. The *Inference Development* phase contains all the different types of assumptions that can be produced after the *Analysis* phase. They may be hypothetical, they may concern current or future outcomes, but they can also be concrete and conclusive. Lastly, the *Dissemination* phase concerns the publication of the investigators' findings in electronic or other type of sources.

Rogers' Behavioural Evidence Analysis Model

M. K. Rogers [Rogers, 2016] introduced a BEA model accustomed to digital investigations. Despite the fact that the model has a linear representation, swapping between phases is applicable. It consists of six phases, namely: *Classification, Context Analysis, Collection, Statistical Analysis, Visualization* and *Decision/Opinion*. *Classification* corresponds to the criminal case selection and the definition of its attributes. *Context Analysis* provides a better understanding of the system under investigation, potential evidence locations and raises the investigator‘s awareness for the existence of anti–forensic techniques. Similarly to aforementioned models, the *Collection* phase is related to "evidence collection and storage in a format that can be analyzed for patterns, linkages, and timeline analyses" [Rogers, 2016]. *Statistical Analysis* comprises the methods used in order to detect potentially abnormal occurrences among the evidence. One of the most common techniques used is frequency analysis, but that does not prohibit the investigators from using a broader spectrum of methodologies. *Visualization* is responsible for presenting the key findings in a timeline manner, whereas *Decision/Opinion* produces a final report that contains the conclusions of the evidence analysis.

The Digital Forensic Intelligence Analysis Cycle (DFIAC)

Quick and Choo [Quick and Choo, 2017] created a model inspired by the UNODC [UNODC, 2011] guidelines. DFIAC is rather similar to The Intelligence Cycle. It contains an additional *Prepare* phase, that is responsible for the contextualization of the crime case, a *Collect, Preserve and Collate* phase, which is a variant of the UNODC‘s equivalent *Collection* with additional pre-processing and its last phase, *Identification of Future Tasks*, an extended version of the *Dissemination* process, including concerns that are yet to be resolved.

It is noticeable that the more concurrent models do not follow a static representation, but have a rather dynamic character. They support internal loops whenever the investigation reveals a new or recurring unresolved concept. Thus, they are rendered more flexible and they provide a higher degree of liberty. The next subsection presents some significant related work in the field of digital criminal profiling, inspired by the aforementioned models.

2.4.2 Related Work

There is a considerable amount of research papers that employ intelligent computing in order to use digital criminal profiling techniques. The great majority of them makes use of demographic data and qualitative basis in order to proceed to inference. Additionally, their main goal is the creation or the validity acknowledgement of an already existing knowledge base. However, contrary to the current work, they are limited to the profile definition and validation and only a few proceed to suspicious behaviour identification. This section presents the related work contributions in the area and the methodology they adopt.

Rogers [Rogers, 2003] made an introduction to the association between criminal profiling and Computer Forensics (CF) by matching cybercriminal Modus Operandi (MO) to activities involving digital assets in a theoretical level. However, the proposal lacked an implementation methodology and results.

One of the first more evolved attempts towards intelligent criminal profiling was the NNPCP project [Strano, 2004], which comprised the creation of a NN capable of performing criminal profiling among different crime types. Its data pool of inputs were official criminal records from the Italian police force. Despite the relatively impressive description for the time the paper was written, the description of the NN architecture and functionalities is rather abstract and no further details on produced results are provided.

Ferrari et al. used Bayesian and Feed-Forward NNs so as to “model criminal behaviour from post-mortem databases of single-victim homicides” [Ferrari et al., 2008] and compared the results of both solutions. The systems used as inputs different psychosocial factors concerning the offenders’ character, as well as the way each crime was conducted and classified different criminal actions to different outputs.

Kwan et al. [Kwan et al., 2008] implemented a Hybrid Intrusion Detection System (HIDS) based on honeypots, with additional cybercriminal-cybercrime association and profiling capabilities. The identification mechanism was based on four metrics; host and network breadth, host and service depth, vulnerabilities and tools. Host breadth is defined by the observation of the connections that derive from the offender and target various ports of the victim system, whereas network breadth is related to the analysis of connections that also come from the offender, but target various network hosts instead. Host depth measures the amount of the offender’s infiltration to the host, while service depth refers exclusively to a particular service every time. Vulnerabilities are the weak points of every system and they can be used so as to identify an attacker’s MO and level of sophistication. Finally, tools are different malicious entities that an offender may employ. All the aforementioned metrics combined define a cybercriminal profile. Once an attack is identified, its metrics are compared to already existing cybercriminal profiles by a similarity index and if no former correlation exists, the attack is categorized as a new profile. Despite the fact that the description of both the system and the experimental setup was complete, no concrete results were presented.

Enache et al. [Enache et al., 2010] designed a multilayer NN that aimed to create a demographical and activity-based cybercriminal profile according to an input set of crime types and their associated activities. The authors claim that some successful associations were made; however, complete results were not presented.

The paper by Islam and Verma [Islam and Verma, 2012] used Fuzzy Logic concepts in order to perform a risk assessment on messages exchanged by various entities in a 3G network, depending on their identity and motives. The system inputs comprised the variable combination of the SMS senders' degree of acquaintance to the device owner and the type of device they have been using. The system output was the overall risk per input combination, measured in a scale from zero to five. Zero represented the lowest degree of risk per SMS, whereas five represented the highest. However, the authors did not present any experiments on actual or simulated data.

Lai et al. introduced "a conceptual framework for profiling internet pirates" [Lai et al., 2013], according to their behavioural traits on technology use. The authors constructed the internet pirate' s profile based on three pillars: "the facts, the behavioural characteristics and the personality particulars" [Lai et al., 2013]. The facts category referred to an amount of various observations inferred by the existing data, such as timestamps and exchanged file types. The behavioural characteristics group incorporated traits concerning a pirate's Internet usage, whereas the personality particulars category comprised more abstract notions, such as personality characteristics, reasons that led to piracy and influences that formed the potential pirate's profile. Afterwards, they created and distributed a questionnaire consisting of content related to the three aforementioned categories. They used the Multidimensional Scaling (MDS) [Borg and Groenen, 2005] methodology, so as to form clusters with correlated characteristics and create the respective piracy profiles.

A methodology schema for cyberstalkers' profiling was suggested by Silde and Angelopulou [Silde and Angelopoulou, 2014]. The authors constructed a culprit's digital profile digital profile, according to various MO characteristics encounter in sociology, law and psychology sources. Moreover, they simulated a cyberstalking scenario between two Virtual Machines (VMs) acting as the victim's and the offender's Windows 7 Personal Computers (PCs). After performing forensic acquisition in both computers, they enumerated the artefacts that could pinpoint to illegal activity, such as e-mail messages, chat segments, etc. and their respective location in the systems. One of the downsides of the methodology was that the artefacts' identification procedure was conducted manually. However, it provided some useful insights for future implementation of mechanisms able to parse files in the aforementioned locations and conduct the essential evidence analysis.

Andro-AutoPsy is a recently introduced antimalware tool with an innovating attribute. Apart from the information about the malware technical characteristics, the tool uses "similarity matching in malware creator-centric information" [Jang et al., 2015], so as to construct the respective criminal profiles. Information concerning the creator's behaviour is usually encountered in ".smali opcodes, metadata in the AndroidManifest.xml file, as well as in serial numbers of various certificates" [Jang et al., 2015]. Andro-AutoPsy is actually a hybrid detection engine, consisting of a rule-based, behavioural detection module and a classification engine that performs comparisons among already existing malware activities and decides upon the suspicion of a sample.

Al Mutawa et al. [Mutawa et al., 2016] suggested a cyberstalking profiling methodology, based on BEA conducted upon forensically acquired computer data. The authors performed statistical analysis on the datasets and were able to identify different groups of characteristics that built certain cyberstalker profiles.

Quick and Choo [Quick and Choo, 2017] introduced an extensive process model for intelligent MF, named Digital Forensics Analysis Cycle (DFIAC). The model was then applied to a procedure of retrieving information from various mobile devices confiscated by the South Australian Police for the time period between 2000 and 2015. The authors were able to successfully establish association links among different criminal entities.

Table 2.2: Digital criminal profiling papers' characteristics

Author(s)	**Approach**	**Platform**	**Data**	**Type**	**Profiling**	**Susp.**	**Res.**
[Rogers, 2003]	Theoretical	Computer	N/A	Behav.	Manual	No	No
[Strano, 2004]	Implement.	Computer	Demographic	Behav.	Auto	No	No
[Ferrari et al., 2008]	Implement.	N/A	Demographic	Behav.	Auto	No	Yes
[Kwan et al., 2008]	Implement.	Computer	Simulation	Machine	Auto	Yes	No
[Enache et al., 2010]	Implement.	Computer	Demographic	Behav.	Auto	No	No
[Islam and Verma, 2012]	Theoretical	Mobile	Undefined	Behav.	Auto	No	No
[Lai et al., 2013]	Implement.	Computer	Demographic	Behav.	Auto	Yes	Yes
[Silde and Angelopoulou, 2014]	Theoretical	Computer	Simulation	Behav.	Manual	No	No
[Jang et al., 2015]	Implement.	Mobile	Real	Hybrid	Auto	Yes	Yes
[Mutawa et al., 2016]	Theoretical	Computer	Demographic	Behav.	Manual	No	No
[Quick and Choo, 2017]	Implement.	Mobile	Real	Behav.	Auto	No	Yes

It is rather noticeable that the criminal profiling research has matured overtime. Recent works are more sophisticated and concrete. Moreover, they contain a higher amount of experiments and results, thus providing stronger proof. While older research papers in the field used novel methodologies for classification and profiling of data stored on physical means, their successors show a trend towards interaction of electronically generated user data. Table 2.2 summarizes the research conducted in the field of digital criminal profiling. A rather paradoxical observation is that even though most of the research papers presented an implementation, only few of them provided actual results. A reason for this controversy is the fact that despite the existence of data sources, most of the authors only use them in a qualitative manner. Since no unsupervised techniques such as clustering are used, it is difficult to provide a quantitative body of results.

The research papers concerning purely mobile criminal profiling started to appear only after 2012, with the work by Islam and Verma [Islam and Verma, 2012] constituting a theoretical attempt. Data provided by law enforcement organizations were merely used for demographic purposes -i.e. approximating the offenders' age group, gender and motivation- and manual profiling, fact that reveals the weakness of interconnection between the forensic evidence and the behavioural fingerprint. Most of the implementations or the theoretical approaches promise automatic profiling, but such a claim is rather weak when it cannot be argumented by the respective results. Only few of the research papers perform simultaneous suspicious entity identification and they are mainly related to machine-generated activity, such as threat and attack propagation detection.

Table 2.3: Methodology Compilation

Current Methodology	**Related Literature**
Induction Phase	
Use Case Selection	Problem Definition [Mena, 2003], Classification [Rogers, 2016], Tasking [UNODC, 2011], [Quick and Choo, 2017]
MO Definition	Theory [Godwin, 2012], Context Analysis [Rogers, 2016], Prepare [Quick and Choo, 2017]
Dataset Formation	Collection [Mena, 2003], [UNODC, 2011], [Rogers, 2016], Crime Scene Data [Godwin, 2012], Collect, Preserve, Collate [Quick and Choo, 2017]
Variable Definition	Feature Selection [Mena, 2003], Evaluation [UNODC, 2011], [Quick and Choo, 2017]
Pre–processing	Information Matching [Mena, 2003], Collation [UNODC, 2011], Collect, Preserve, Collate [Quick and Choo, 2017]
Ground Truth Generation	Building Digital Profile [Mena, 2003], Inference Development [UNODC, 2011], Hypotheses [Godwin, 2012]
Investigation Phase	
Application	Building Digital Profile [Mena, 2003], Empirical Analysis [Godwin, 2012], Statistical Analysis [Rogers, 2016]
Evaluation	Evaluate New Profile Validity [Mena, 2003], Visualization[Rogers, 2016]
Testing on Unknown Data	Evaluate New Profile Validity [Mena, 2003]
Selection	Decision/Opinion [Rogers, 2016], Future Tasks Identified[Quick and Choo, 2017]

In terms of applied digital criminal profiling methodologies, the authors do not hesitate to combine more than one methodologies and use interdisciplinary notions, so as to achieve better results.

Table 2.3 shows the association between the work presented in the current thesis and the aforementioned research papers. The left column comprises the steps of the current methodology, whereas its right equivalent is a part of the already existing models. While the similarity level between the components of the Induction phase and the related literature is considerably high, the Identification phase equivalents cover broader and different scopes, but have similar high level representation.

The following section presents the digital criminal profiling and suspicious pattern identification methodology of this thesis. Inspired by the methodology compound trend, we present a concatenation of the most important digital criminal profiling phases and fill the research gap by performing suspicious pattern identification for evidence related to offenders' behaviour.

2.5 Proposed Methodology

As already mentioned in the previous sections, research papers on digital criminal profiling and behavioural analysis, in addition to their multi–disciplinary nature, they also tend to adopt and adapt elements from many different methodologies in order to improve the overall quality levels of their outcomes. The methodology presented in this section combines the digital criminal profiling techniques with newly introduced suspicious pattern identification capabilities. It comprises two main phases, Induction and Identification. The Induction phase is related to the construction of a culpable profile and the respective data management. The Identification phase consists

Table 2.4: Methodology Phases

Induction	**Identification**
Use Case Definition	Application
Dataset Selection	Evaluation
Pre-processing	Testing on Unknown Data
Ground Truth Generation	Selection

of the Fuzzy Systems, NN (plain and pattern recognition perceptrons) and ANFIS training and validation, their evaluation and their behaviour testing on previously unknown data. Table 2.4 depicts the content of the two phases.

The methodology presented in this thesis is a concatenation of previously adopted criminal profiling models, enhanced by a suspicious pattern identification routine. The next paragraphs explain each phase in detail.

Use Case Definition

Before any profiling or identification activity takes place, the object of investigation has to be properly defined. The *Use case definition* phase comprises the selection of a criminal activity, the study of the digital fingerprint the offender leaves behind according to insight from previous cases, and the definition of a MO which can be later modeled into relationships between various data and metadata attributes.

Dataset Selection

Different use cases have different dataset requirements. Usually, criminal profiling methodologies mainly target user generated data and metadata (See Subsection 2.2.1.2). However, application and OS generated data can also have significant value. A requirement for the conduction of a successful identification process is the sufficient quantity of patterns in the candidate datasets, which can be met with samples of prolonged mobile usage or simulation.

Pre-processing

Even though datasets can be rich in content, not every piece of information suits the investigation needs. *Pre-processing* is responsible for defining the most crucial attributes and giving them the appropriate format for the application of quantitative methodologies.

Ground Truth Generation

Suspicious pattern identification techniques require an evaluation phase, where the calculated results have to be compared to an initial target value, known as the ground truth in order to assess the methods' efficiency. In the *Ground truth generation* phase, each pattern receives a designated value of suspiciousness between five discrete values, according to its resemblance to the criminal MO characteristics.

Application

Once the *Pre-processing* and *Ground truth generation* phases are complete, the suspicious pattern identification techniques can be applied. For the current thesis, this phase includes the configuration of Fuzzy Systems and the training of two NN perceptrons and ANFIS. Once the results are generated, the evaluation phase is ready to begin.

Evaluation

During this phase, the generated results from the *Application* phase are compared to the ground truth and performance metrics such as accuracy, precision and recall are calculated for each suspiciousness category. The next phase, namely *Testing on unknown data* is applicable only for the NNs and ANFIS, since Fuzzy Systems have no memory or learning capabilities.

Testing on Unknown Data

The particular phase is optional and can only be applied to techniques that support learning features, such as NNs. The techniques are applied anew to a set of previously unknown data to the system, that can either be parts of the initial dataset or acquired directly from test devices. Once it is complete, its evaluation results are compared to the evaluation results of the application phase and previous assumptions are either verified or rejected.

Selection

The *Selection* phase can either occur directly after the evaluation phase, or after the *Testing on unknown data* phase. In the first case, the results generated during the application phase are observed and the most appropriate alternative is selected. In the second case, the performance of the system is cross-evaluated and if the performance declining is considerable, the experiments are repeated. The phase ends with the selection of the best performing setup.

The proposed methodology will be applied, evaluated and tested in the next two chapters, with the use of Fuzzy Systems, NNs and ANFIS. The chapter related to Fuzzy Systems will examine the case of intentionally illegal usage conducted by PPDR agents, whereas the chapter related to NNs and ANFIS will examine traditional criminal investigation cases, specifically the digital fingerprints of cyberbullies and low-level drug dealers.

2.6 Summary of the Chapter

The current chapter served a double scope. Firstly, it provided a general background of the MF field, its basic principles and research directions, while identifying promising research directions that among others included methodologies related to MFDA and more specifically to digital criminal profiling. Secondly, a State-of-the-Art (SoA) analysis of the existing digital criminal profiling methodologies was performed. The study of the field led to the discovery of research gaps for a concrete methodology incorporating both digital criminal profiling and suspicious activity identification. The chapter concluded with the presentation of the proposed methodology, which will be elaborated in the rest of the thesis.

The outcome of this chapter comprises the following publication:

- Konstantia Barmpatsalou, Tiago Cruz, Edmundo Monteiro, and Paulo Simoes. 2018. Current and Future Trends in Mobile Device Forensics: A Survey. *ACM Computing Surveys* 51, 3, 1–31. Impact factor: 6.74

Chapter 3

Fuzzy Systems for Suspicious Pattern Identification

My crystal ball is fuzzy.

(Lotfi A. Zadeh)

Evidence associated to human behaviour show high levels of uncertainty [Gegov, 2010] and thus methods such as Fuzzy Systems, that offer the variety of multiple outcomes are considered a suitable approach. This chapter constitutes a preliminary proof of concept, which aims to show that Fuzzy Logic can efficiently classify SMS patterns from mobile devices into different groups of suspiciousness. The related background and methodology steps are elaborated in detail within the following sections.

3.1 Fuzzy Systems

Fuzzy Systems can “model human reasoning from imprecise and incomplete information by giving definitions to vague terms and allowing construction of a rule base” [Zadeh, 1965]. L. A. Zadeh introduced the concept of Fuzzy Logic by claiming that that bivariate logic is not sufficient for solving complex problems. Moreover, he asserted that “as the complexity of a system increases, the ability to make precise and significant statements about its behaviour diminishes until a threshold is reached, beyond which precision and significance become almost mutually exclusive characteristics” [Zadeh, 1973]. Contrary to Traditional Logic approaches, where a statement can be declared as either true (0) or false (1), Fuzzy Logic can theoretically support infinite intermediate conditions, i.e. approximations.

Some of the examples of Fuzzy states include but are not limited to variables such as temperature (high-medium-low), height (tall-average-short), performance (excellent-very good-fair-fail) and others, that can simultaneously receive quantitative (numerical) and qualitative (linguistic) characteristics. Moreover, “Fuzzy Logic can be seen as a reasoning formalism of humans where all truths are partial or approximate and any falseness is represented by partial truth” [Siddique and Adeli, 2013]. The following subsections dive deeper into the concept of Fuzzy Logic, Systems and their principles.

3.1.1 Fuzzy Membership Functions

A membership function $\mu_F(x)$ of a fuzzy set F in a universe of discourse X matches every x to any real number within the [0,1] interval. F is formally depicted as a pair of the variable x and its representation via $\mu_F(x)$.

$$F = \{x, \mu_F(x) | x \in X\} \tag{3.1}$$

In Traditional Logic, the membership function receives strictly two values, either $\mu_F(x) = 0$ or $\mu_F(x) = 1$ The most widespread membership function types are the Triangular, Trapezoidal, Bell, Gauss and Gauss2. However, other custom functions can be used without particular limitations.

Triangular Membership Function

The Triangular membership function is the three-point set $\{\alpha, \beta, \gamma\}$ with $\alpha < \beta < \gamma$. Its definition is provided in Equation 3.2 and its diagram in Figure 3.1.

$$\mu_F(x) = \begin{cases} 0 & , x < \alpha \\ \dfrac{x-\alpha}{\beta-\alpha} & , \alpha \leq x \leq \beta \\ \dfrac{\gamma-x}{\gamma-\beta} & , \beta \leq x \leq \gamma \\ 0 & , \gamma \leq x \end{cases} \tag{3.2}$$

The aforementioned points are the coordinates that form the triangular shape of the function. More precisely, β is the peak point of the triangle, whereas α and γ create its base.

Trapezoidal Membership Function

The parameter set of points $\{\alpha, \beta, \gamma, \delta\}$ with $\alpha < \beta < \gamma < \delta$ creates the trapeze of the homonym membership function. The top side incorporates the points β

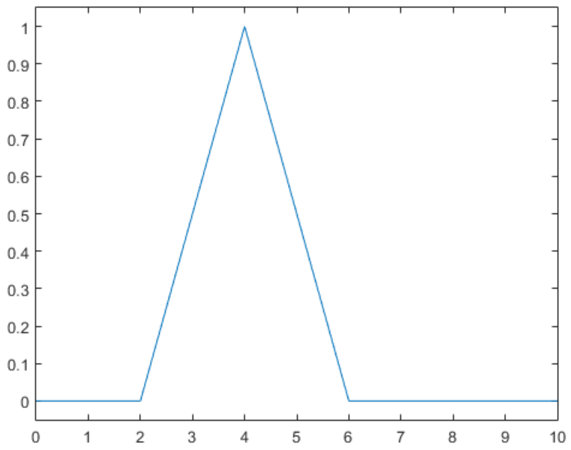

Figure 3.1: Triangular membership function

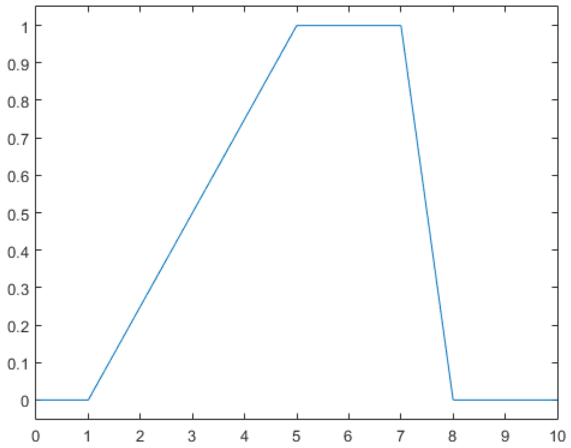

Figure 3.2: Trapezoidal membership function

and γ, while the base consists of the points α and δ. Equation 3.3 and Figure 3.2 show its mathematical and spatial definition.

$$\mu_F(x) = \begin{cases} 0 & , x < \alpha \\ \dfrac{x-\alpha}{\beta-\alpha} & , \alpha \leq x \leq \beta \\ 1 & , \beta \leq x \leq \gamma \\ \dfrac{\delta - x}{\delta - \gamma} & , \gamma \leq x \leq \delta \\ 0 & , \delta \leq x \end{cases} \tag{3.3}$$

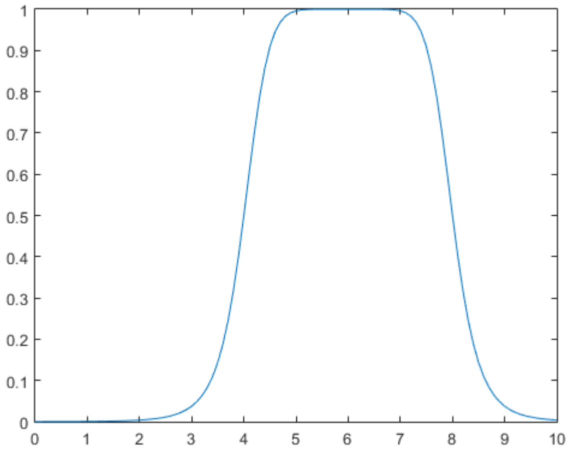

Figure 3.3: Bell membership function

Bell Membership Function

The format of the Bell membership function is presented in Equation 3.4. The parameter γ defines the center of the Bell curve, α defines its width whereas β defines its slope and is usually a number greater than zero. A representation of the Bell membership function is provided in Figure 3.3.

$$\mu_F(x) = \frac{1}{1 + \left|\dfrac{x-\gamma}{a}\right|^{2\beta}} \tag{3.4}$$

Gauss Membership Function

Equation 3.5 shows the formula of the Gauss membership function. The parameter γ controls the center of the curve, whereas σ^2 is the respective variance, a means of defining the curve width. Figure 3.4 depicts the Gaussian curve.

$$\mu_F(x) = e^{\dfrac{-(x-\gamma)^2}{2\sigma^2}} \tag{3.5}$$

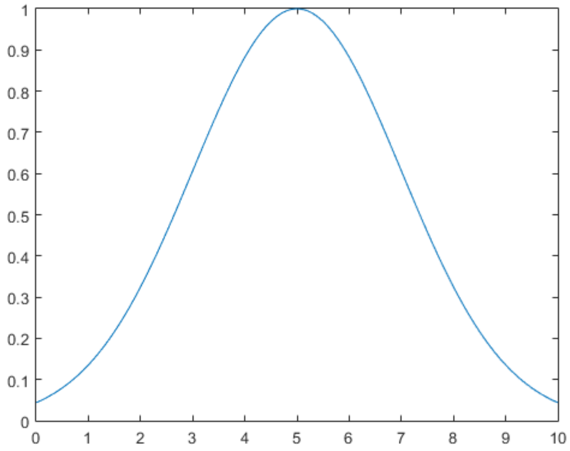

Figure 3.4: Gauss membership function

Gauss2 Membership Function

The Gauss2 membership function is a composite Gaussian function, consisting of two parameter sets, $\{\sigma_1, \gamma_1\}$ and $\{\sigma_2, \gamma_2\}$. The function represented by Equation 3.6 is responsible for defining the shape of the leftmost curve, whereas the respective function in Equation 3.7 controls the shape of the rightmost one.

$$\mu_F 1(x) = \begin{cases} e^{\dfrac{-(x-\gamma_1)^2}{2\sigma_1^2}} & , x \leq \gamma_1 \\ 1 & , otherwise \end{cases} \tag{3.6}$$

$$\mu_F 2(x) = \begin{cases} 1 & , x \leq \gamma_2 \\ e^{\dfrac{-(x-\gamma_2)^2}{2\sigma_2^2}} & , otherwise \end{cases} \tag{3.7}$$

Figure 3.5 presents two Gauss2 membership functions with the same σ_1 and σ_2 parameters, but with different γ combinations. For the upper curve the γ_1 value is less than γ_2, with $\{\gamma_1, \gamma_2\} = \{4, 8\}$, while the parameter values are inverse for the bottom curve, with $\{\gamma_1, \gamma_2\} = \{8, 4\}$. Generally, when γ_1 is less than γ_2, the maximum value of the function spikes up to 1. Otherwise, it is always less than the aforementioned value.

Other membership function types, such as the Π-shaped,the Z-shaped and the Sigmoidal are also widely used. However, their elaboration is omitted because they were not used in the current thesis.

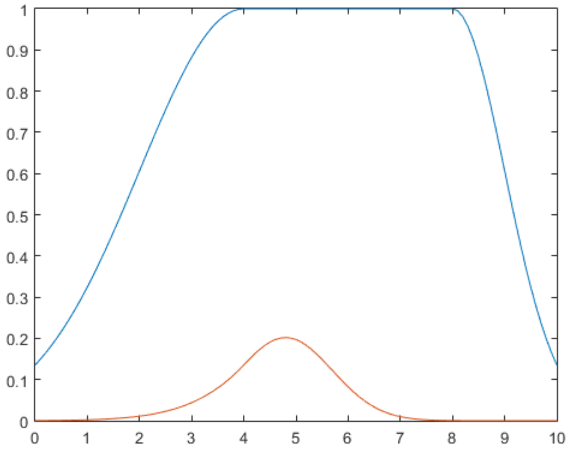

Figure 3.5: Gauss2 membership function

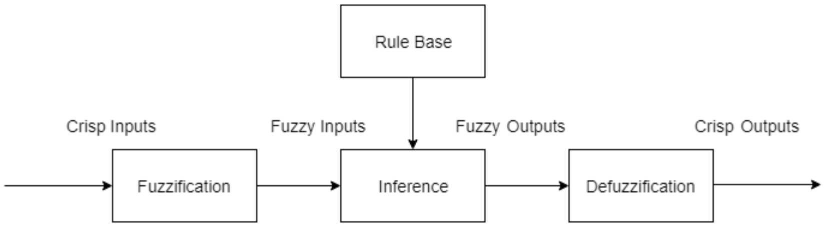

Figure 3.6: Fuzzy System architecture, adapted from Kumar and Verma [Kumar and Verma, 2015] and Abraham [Abraham, 2005]

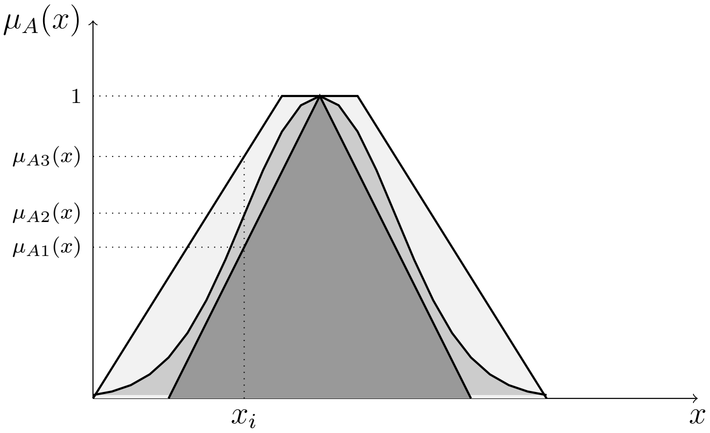

Figure 3.7: Fuzzification for different membership function types, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

3.1.2 The Fuzzy Inference Mechanism

Figure 3.6 depicts the generic form of a Fuzzy System architecture. The system consists of the following parts: the Rule Base, which contains antecedents and consequents and is responsible for forming the IF-THEN rules that define the scope and functionality of the system; the Fuzzification procedure, that converts the crisp inputs into fuzzy with the aid of the membership functions; the Inference that “is responsible for the decision-making procedure” [Abraham, 2005]; and the Defuzzification, which converts the fuzzy outputs to their crisp equivalents.

Fuzzification

As mentioned in the beginning of this subsection, the fuzzification procedure uses the membership functions presented in Subsection 3.1.1, so as to map a certain crisp input x_i to its fuzzy format, $\mu_{An}(x_i)$. A fuzzification example is provided in Figure 3.7, where $\mu_{A1}(x_i)$, $\mu_{A2}(x_i)$ and $\mu_{A3}(x_i)$ are fuzzified inputs corresponding to a trapezoidal, a triangular and a Gauss membership function.

Defuzzification

Defuzzification is the inverse of the fuzzification process, where the fuzzy output, calculated by the Inference module, is transformed anew into its crisp format. The next paragraphs describe the various defuzzification methods encountered in literature. There are no standardizations or specific guidelines towards the selection of an appropriate defuzzification method and the whole procedure is rather problem-centric.

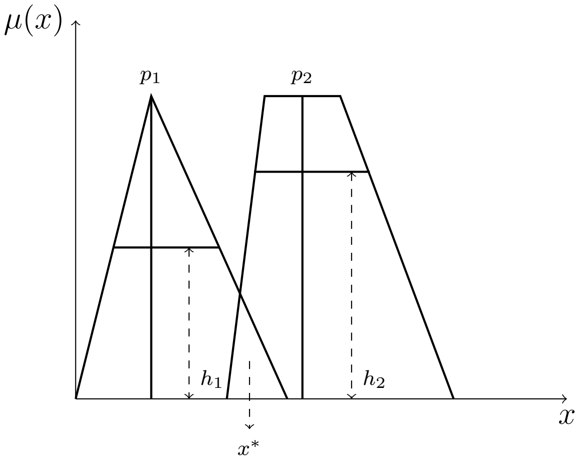

Figure 3.8: Max-membership defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

Max-Membership

Alternatively known as the height method, max-membership constructs the weighted sum of the peak values p_n of all the fuzzy sets existing in the universe of discourse. Its calculation is provided in Equation 3.8, where h_n is the fuzzy sets height.

$$x^* = \frac{\sum_{n=1}^{m} p_n h_n}{\sum_{n=1}^{m} h_n} \tag{3.8}$$

Centre of Gravity

The centre of gravity method is also known as centre of area or centroid method and it is the most common defuzzification approach in the relative literature [Siddique and Adeli, 2013]. The method calculates the centroid of the membership function $\mu(x)$ curve. For a continuous universe, the calculation manner is provided in Equation 3.9.

$$x^* = \frac{\int x\mu(x)dx}{\int \mu(x)dx} \tag{3.9}$$

For a discrete universe, the Equation above is transformed in the following manner:

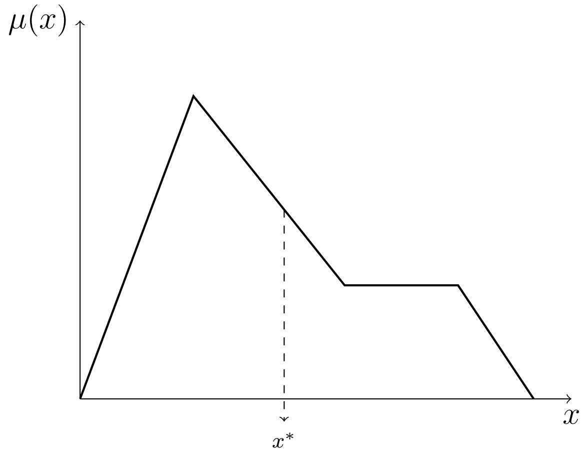

Figure 3.9: Centre of gravity defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

$$x^* = \frac{\sum_{n=1}^{m} \mu(x)x_n}{\sum_{n=1}^{m} \mu(x)} \tag{3.10}$$

Weighted Average

The weighted average method is mostly appropriate for symmetric membership functions and its expression is presented in Equation 3.11.

$$x^* = \frac{\sum \mu(x)x`}{\sum \mu(x)} \tag{3.11}$$

Mean-Max Membership

This method is often reported as the middle of maxima and is calculated as “the mean of all the local maxima” [Siddique and Adeli, 2013].

$$x^* = \frac{\sum_{n=1}^{m} \mu_{max}(x_n)}{m} \tag{3.12}$$

There are many different forms of defuzzification and certain Fuzzy System types use specific variations. For example, the Takagi-Sugeno Fuzzy Systems use the weighted average method, whereas Mamdani Fuzzy Systems use the centre of

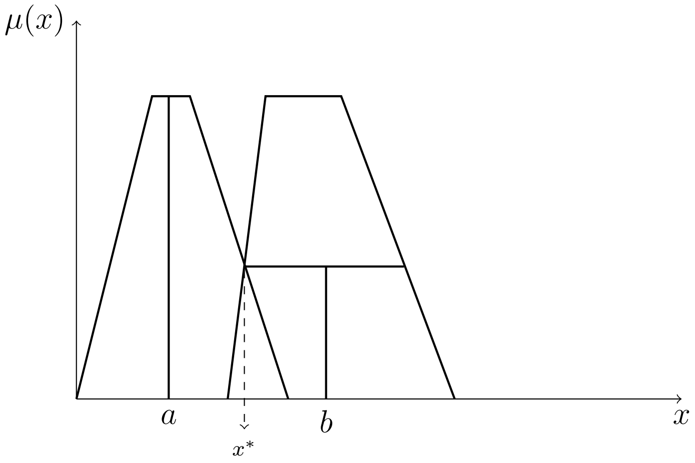

Figure 3.10: Weighted average defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

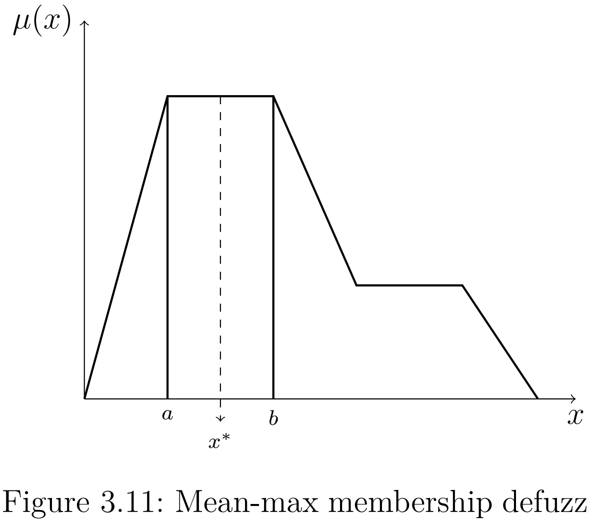

Figure 3.11: Mean-max membership defuzzification, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

gravity technique. More details about each system type can be encountered in the following section.

3.1.3 Fuzzy System Types

This subsection presents the two more widespread Fuzzy System types, Mamdani and Takagi Sugeno, which will be used in the current thesis. While the former will be used as standalone mechanisms, the latter constitute a structural part of the Adaptive Neuro-Fuzzy Inference System (ANFIS).

3.1.3.1 Mamdani Fuzzy Systems

A Mamdani prototype was initially introduced as "an attempt to control a steam engine and boiler using a set of linguistic control rules obtained from an experienced human operator" [Mamdani, 1974]. The simplest form comprises two inputs (x_1, x_2) of M and N value range, and one output y of O value range. Consequently, the input x_1 consists of k=1...P membership functions, whereas the input x_2 and the output y consist of l=1...Q and i=1...S membership functions. The relationship that forms a rule R_n is given by Equation 3.13.

$$R_n\text{: } \mathit{IF}\ x_1 == M_k\ \mathit{AND}\ x_2 == N_l\ \mathit{THEN}\ y == O_i \tag{3.13}$$

"The rules connect the input variables with the output variables and are based on the fuzzy state description that is obtained by the definition of the linguistic variables" [Zimmermann, 1996]. The upper bound for the number of rules is provided by the relationship $R_n \subset P \times Q$. The most common method of the rules' firing strength calculation for Mamdani Fuzzy Systems is max/min. More precisely, the minimum firing strength is selected as the preferred output in a fuzzy set format. Another, less common approach is max/product, where the output is defined by the algebraic product of the input membership functions upon modification.

Mamdani Fuzzy systems follow intuitive variable configuration and output inference, fact that renders them more comprehensive for users with no previous experience in the field, because they do not have to be aware of the dynamic equation that describes the system [Kumar and Verma, 2015]. As a result, they are mostly suitable for decision support problems [Blej and Azizi, 2016]. The membership function tweaking procedure is rather linear and straightforward. Moreover, due to their ease-of-use, they are suitable for both *Multi-Input Multi-Output (MIMO)* and *Multi-Input Single-Output (MISO)* models [Lilly, 2010]. Lastly, the output of Mamdani systems is also a fuzzy set, with a respective membership function.

3.1.3.2 Takagi-Sugeno Fuzzy Systems

Takagi-Sugeno systems were introduced by Takagi, Sugeno and Kang as a means of "generating fuzzy rules from a given input/output data set" [Takagi and Sugeno, 1993]. The simplest representation form consists of two inputs (x_1, x_2) and one output y. The input format is the same as in Mamdani Fuzzy Systems, where the input x_1 comprises k=1...P membership functions, whereas the input x_2 comprises l=1...Q membership functions. The output has an entirely different format and is a crisp and usually polynomial function $y = f(x_1, x_2)$. There are no restrictions to the potential output function if the relationships entirely cover the span of the model to be examined. The relationship that forms a rule R_n is provided by Equation 3.14.

$$R_n\textit{: IF } x_1 == M_k \textit{ AND } x_2 == N_l \textit{ THEN } y == f(x_1, x_2) \tag{3.14}$$

Given the Equation 3.14, the output function format for a random rule will be the following: $y_k = m_k x_1 + n_k x_2$.

Contrary to the Mamdani equivalent, the Takagi-Sugeno output is a continuous linear function, thus offering higher levels of "accuracy and overall computational efficiency and making its model an appropriate candidate for functional analysis" [Kumar and Verma, 2015]. In Takagi-Sugeno models, the defuzzification procedure is performed by the weighted average method and the computational duration is significantly lower [Siddique and Adeli, 2013]. Takagi-Sugeno systems are also more flexible and "they are more easily integrated into adaptive techniques, such as Neuro-Fuzzy Systems" [Lilly, 2010]. Lastly, Takagi-Sugeno systems are more suitable for solving "dynamic and non-linear" problems [Blej and Azizi, 2016].

Evidence associated to human behaviour shows high levels of uncertainty [Gegov, 2010] and thus intelligent computation methods are considered a suitable approach. This chapter constitutes a preliminary proof of concept, which aims to show that Fuzzy Logic can efficiently classify SMS patterns from mobile devices into different groups of suspiciousness. The steps of the respective methodology include the acquisition of expert knowledge, rule inference, definition of datasets and variables, membership function selection and evaluation. They are elaborated in detail within the following sections.

3.2 Use Case Definition

Conducting a research correlated to actual criminal investigation would be impossible without prior expert knowledge available. The knowledge required for the construction of the fuzzy rule base is a hybrid compilation of incidents the use cases provided in the SALUS FP7 Project deliverables [SALUS, 2014] and of on-field investigation practices provided by an officer of the Greek Police Escort Teams Department (GPETD) [Barmpatsalou et al., 2018b]. After collecting all the essential insights, the rules of each Fuzzy System were structured and presented. The use case of PPDR officers infiltrating a protesters' group will be examined by the investigation of SMS from three different devices.

Another challenge that was faced concerned the lack or unavailability of actual evidence retrieved from devices involved in criminal activities. As a result, delinquent actions had to be simulated and injected in the datasets as standalone patterns. The a-priori expert knowledge served as a solid background for the rule generation, which is analyzed in the following subsection.

3.2.1 Rule Inference

With the use of the expert knowledge mentioned in the previous section, respective rules concerning the data categories were created for the use case. The rules were formed from a combination of the available data and the investigation directives for the use case. If the use case changes, the rules are as well altered. For the scenario of the infiltration by PPDR officers, the following setup was created.

Sent SMS texts retrieved from a device of a potential infiltrator may have the following attributes:

- If officers are infiltrators, they will use their devices to communicate with their accomplices only in cases of extreme necessity. As a result, the rate with which a sent message will appear is going to be very low.
- Most of the accomplices may use one-time payphones, which are equipped with Subscriber Identity Modules (SIMs) from the same country the incidents occur. Recipients with local numbers are considered more suspicious.
- According to the GPETD experts, messages exchanged during rioting or right before similar incidents are very short in length.
- As a result, the sent SMS pattern with the combination (very low appearance frequency–very short length-local country code source) is considered the most suspicious.

Nonetheless, the rule inference procedure needs a functioning dataset that is able to fulfill the research requirements in size and content. The following subsection covers in detail the challenges in the quest for a suitable data source.

3.3 Dataset Selection

It is a generally accepted truth that "data are barely shared among the Digital Forensics community" [Grajeda et al., 2017]. Law enforcement agencies have adopted a strict policy about sharing on-field acquired data with the scientific community, so the majority of the publicly available datasets are results of human or computer generated simulations. Digital Forensic datasets are scattered and derive from a variety of sources. Disk images and network dumps are significantly higher in quantity than mobile device images or parts of the device storage. Most of the existing datasets related to mobile devices, such as Computer Forensic Reference Data Sets (CFReDS) [National Institute of Standards and Technology (NIST), 2016] and Digital Corpora [Digital Corpora, 2017] correspond either to devices with older mobile Operating System (OS) versions or the available data are not enough in quantity so as to adhere to scenarios that demand repetition of experiments over time. Moreover, datasets relevant to actual criminal investigations are not made publicly available to the community and are under strict authorities' jurisdiction.

The CDA [Wagner et al., 2014] dataset is a collection of Android data and metadata items from various users worldwide, who voluntarily provide them by installing an application-agent. Access to the full dataset is provided after signing a mutual agreement, where one of the ends is either an academic entity or an organization. The data are sorted by the device they were acquired from and each part contains information collected over a period of six months. Additionally, they are stored in a Comma Separated Value (.csv) file and are formatted according to a scheme containing a unique numeric identifier, a timestamp, a label of the data type and a string field of the corresponding attributes. A more detailed representation of a data tuple can be found below:

$$\texttt{tuple} = \{\texttt{id.}, \texttt{timestamp}, \texttt{type}, [\texttt{attr.}_1, \texttt{attr.}_2, .., \texttt{attr.}_\texttt{n}]\} \tag{3.15}$$

The dataset used comprises data from three separate devices. The data tuple in Equation 3.15 is split in such a way that each column represents a unique attribute. Not every piece of information is useful for the research, so data are filtered and redundancies are removed. The data type used is SMS and it has the following three respective attributes, which are described analytically in the next section.

$$\texttt{SMS}(\texttt{name}, \texttt{length}, \texttt{location}) \tag{3.16}$$

Table 3.1: Fuzzy variable ranges

Input Variable	Fuzzy Approximation	Numerical Range
Length	VERY SHORT, SHORT, MEDIUM, LONG, VERY LONG	1-600 characters
Appearance Frequency	VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH	1-1100 appearances
Country Code	FOREIGN, UNDEFINED, LOCAL	0, 1 and 2

3.4 Pre-Processing

Not all of the data inputs of Equation 3.15 in their raw format constitute Fuzzy-Logic ready material. Therefore, some quantification pre-processing has to take place for the data to constitute valid fuzzy inputs.

Name:

In the CDA Dataset, the *Name* attribute corresponds to series of anonymized entities that are represented by an alphanumeric string. The feature by itself does not have a significant research value for the current thesis, but if the number of encounters per name are enumerated, the *Appearance Frequency* variable is created. *Appearance Frequency* signifies the number a certain name appeared as an SMS interaction with the device owner over a specific period of time.

Length:

Length is the attribute that remained intact, without any need for additional preprocessing. It is a continuous positive integer that corresponds to the total number of characters a SMS text consists of.

Location:

The *Location* attribute is represented by linguistic terms in the CDA dataset and refers to whether the phone number of an entity that interacted with the device owner is foreign, local and unknown or undefined, due to parsing errors. The generated *Country Code* variable has three discrete values that are presented in Table 3.1.

Afterwards, during the system design phase, the criteria for “readability and interpretability of the variables and the rules that are deriving from them” [Lima and Camargo, 2014], as they were presented in the papers by Guillaume and Charnomordic [Guillaume and Charnomordic, 2012] and Gacto et al. [Gacto et al., 2011] were reviewed and verified.

- While aiming to maintain a high degree of semantic cohesion, every fuzzy set should represent a well-defined and non-vague concept. The fuzzy sets and the value range of each variable participating in the current research have specific meanings; fact that can be proven by consulting Table 3.1.

- Each fuzzy variable should neither exceed 9, nor deceed 5 range fields, which is defined as the threshold for human perception capabilities [Lima and Camargo, 2014]. In the current experiment, the maximum number of different value ranges is 5, number that falls between the aforementioned limit.

- There is no point within the system's universe of discourse that does not belong to at least one fuzzy set.

- A fuzzy set should be normal; in a fuzzy system $\overline{F}$, there should always exist at least one χ, the membership degree (height) of which should be equal to 1.

- It is obligatory that "all fuzzy sets should overlap in a certain degree" [Lima and Camargo, 2014].

Once all the requirements for the variables are met and before the system evaluation begins, the ground truth is generated according to the existing expert knowledge.

3.5 Ground Truth Generation

In an earlier paper [Barmpatsalou et al., 2018b] concerning the work included in the current chapter, we introduced an alternative representation of the output suspiciousness. Instead of using the classic binary format (0: not suspicious - 1: suspicious), the output is a fuzzy variable, receiving values within the [0,1] interval. Values closer to zero are considered innocent, whereas values closer to one are regarded as more suspicious. Despite the fact that the output can receive any number within the aforementioned interval, five representative values (0.15, 0.25, 0.5, 0.75, 1) were indicated as thresholds for each suspiciousness category. Table 3.2 demonstrates the assignment and the respective linguistic values. Moreover, each fuzzy rule deriving from every variable combination was given a suspicious linguistic value. Table 3.3 presents the generated suspiciousness per rule.

Lastly, manual assignment of ground truth labels was employed per pattern in the dataset, according to the principles generated by Tables 3.1, 3.2 and 3.3. Once the process was completed, the evaluation phase was ready to begin.

Table 3.2: Fuzzy suspiciousness values, adapted from Barmpatsalou et al. [Barmpatsalou et al., 2018b]

Value	Suspiciousness Level
0.15	Very Low
0.25	Low
0.5	Medium
0.75	High
1	Very High

3.6 Evaluation

The Fuzzy System evaluation and simultaneous membership function selection was a rather complicated procedure. In order to select the appropriate setup for each dataset assigned to the respective Fuzzy System, an evaluation methodology based on the comparison of the Fuzzy Systems' output and the ground truth values was employed. With the ground truth considered the target and the fuzzy output being the feature variable, the fuzzy output values of five fuzzy systems configured with different membership functions (Triangular, Trapezoidal, Bell, Gauss and Gauss2) were classified into five different groups of suspiciousness, according to the Ground truth generation section.

Five machine learning algorithms, namely k-Nearest Neighbour (kNN), Support Vector Machine (SVM), Random Forest Classification (RFC), Naive Bayes (NB) and AdaBoost (Ada) are employed in order to classify the fuzzy datasets' outputs in comparison to the ground truth targets and produce results about the performance of each membership function. The sampling technique applied is 10-fold cross-validation.

The aforementioned procedure produces a 5x5 confusion matrix and classification metrics (Accuracy, Precision, Recall, Area Under Curve (AUC) and False Positive Rate (FPR)) are calculated for each category and then presented as an average score. More precisely, Accuracy refers to the ratio of the correctly classified patterns per category and the total number of patterns.

$$Accuracy = \left\langle \frac{TP_c + TN_c}{TP_c + FP_c + FN_c + TN_c} \right\rangle, \quad c \in S_G \tag{3.17}$$

Precision is the amount of True Positive (TP) patterns over the sum of TP and False Positive (FP) values.

$$Precision = \left\langle \frac{TP_c}{TP_c + FP_c} \right\rangle, c \in S_G \tag{3.18}$$

Table 3.3: Ground truth generation per rule

Appearance Frequency	**Length**	**Country Code**	**Suspiciousness Level**
Very Low	Very Short	Foreign	Medium
Very Low	Very Short	Undefined	High
Very Low	Very Short	Local	Very High
Low	Very Short	Foreign	Medium
Low	Very Short	Undefined	High
Low	Very Short	Local	Very High
Medium	Very Short	Foreign	Low
Medium	Very Short	Undefined	Medium
Medium	Very Short	Local	Medium
High	Very Short	Foreign	Very Low
High	Very Short	Undefined	Low
High	Very Short	Local	Low
Very High	Very Short	Foreign	Very Low
Very High	Very Short	Undefined	Very Low
Very High	Very Short	Local	Low
Very Low	Short	Foreign	Medium
Very Low	Short	Undefined	Medium
Very Low	Short	Local	Very High
Low	Short	Foreign	Low
Low	Short	Undefined	Medium
Low	Short	Local	High
Medium	Short	Foreign	Low
Medium	Short	Undefined	Low
Medium	Short	Local	Medium
High	Short	Foreign	Very Low
High	Short	Undefined	Low
High	Short	Local	Medium
Very High	Short	Foreign	Very Low
Very High	Short	Undefined	Very Low
Very High	Short	Local	Low
Very Low	Medium	Foreign	Medium
Very Low	Medium	Undefined	High
Very Low	Medium	Local	Very High
Low	Medium	Foreign	Low
Low	Medium	Undefined	Medium
Low	Medium	Local	High
Medium	Medium	Foreign	Low
Medium	Medium	Undefined	Low
Medium	Medium	Local	Medium
High	Medium	Foreign	Very Low
High	Medium	Undefined	Low
High	Medium	Local	Low
Very High	Medium	Foreign	Very Low
Very High	Medium	Undefined	Very Low
Very High	Medium	Local	Very Low
Very Low	Long	Foreign	Low
Very Low	Long	Undefined	Low
Very Low	Long	Local	Medium
Low	Long	Foreign	Very Low
Low	Long	Undefined	Low
Low	Long	Local	Low
Medium	Long	Foreign	Very Low
Medium	Long	Undefined	Low
Medium	Long	Local	Low
High	Long	Foreign	Very Low
High	Long	Undefined	Very Low
High	Long	Local	Low
Very High	Long	Foreign	Very Low
Very High	Long	Undefined	Very Low
Very High	Long	Local	Very Low
Very Low	Very Long	Foreign	Low
Very Low	Very Long	Undefined	Medium
Very Low	Very Long	Local	Medium
Low	Very Long	Foreign	Low
Low	Very Long	Undefined	Low
Low	Very Long	Local	Medium
Medium	Very Long	Foreign	Very Low
Medium	Very Long	Undefined	Low
Medium	Very Long	Local	Low
High	Very Long	Foreign	Very Low
High	Very Long	Undefined	Very Low
High	Very Long	Local	Low
Very High	Very Long	Foreign	Very Low
Very High	Very Long	Undefined	Very Low
Very High	Very Long	Local	Very Low

Recall is the number that results from the division of the TPs with the total of TP and False Negative (FN) patterns.

$$Recall = \left\langle \frac{TP_c}{TP_c + FN_c} \right\rangle, c \in S_G \tag{3.19}$$

FPR is the ratio of the True Negative (TN) patterns over the total number of negatives.

$$FPR = \left\langle \frac{TN_c}{FP_c + TN_c} \right\rangle, c \in S_G \tag{3.20}$$

Equations 3.17, 3.18, 3.19 and 3.20 describe how each metric is calculated for every class c that belongs to the S_G set of suspiciousness values. Finally, the AUC metric refers to “the higher positive-over-negative value ranking capability of a classifier” [Barmpatsalou et al., 2018b].

3.7 Summary of the Chapter

This chapter presented the basic concepts of the Fuzzy Systems theory, focusing on the concepts used in the current research. Afterwards, the methodology presented in Chapter 2 was applied and each of its steps was further analyzed. Initially, the PPDR agents' riot infiltration use case was presented and their MO concerning the SMS fingerprint was defined. It was then represented in the format of fuzzy rules, so as to be inserted into the fuzzy rule base. The essential data were extracted from the CDA dataset and modified appropriately during the Pre-processing phase. Finally, each MO pattern combination was given a ground truth value and the Evaluation phase was described. The results of the respective Evaluation procedure will be presented in Chapter 5.

The outcomes of this chapter include the following publications:

- Barmpatsalou, K., Cruz, T., Monteiro, E. and Simoes, P. (2017). From fuzziness to criminal investigation: An inference system for Mobile Forensics, in Kambourakis, G., Shabtai, A., Kolias, K. and Damopoulos, D., *Intrusion Detection and Prevention for Mobile Ecosystems*, pages 117-132, CRC Press.
- Barmpatsalou, K., Cruz, T., Monteiro, E. and Simoes, P. (2018). Fuzzy system-based suspicious pattern detection in mobile forensic evidence, in Matousek, P. and Schmiedecker, M., editors, *Digital Forensics and Cyber Crime, 9th International Conference ICDF2C 2017*, pages 106-114, Cham, Springer International Publishing.

Chapter 4

Neural Networks and ANFIS for Suspicious Pattern Identification

'Tis to create, and in creating live.

(Lord Byron)

The previous chapter described the adaptation of the proposed methodology for a scenario that employed the use of Fuzzy Systems. However, their use has some shortcomings, that the current chapter is going to attempt to address. Due to their nature, Fuzzy Systems are incapable of maintaining a memory of previous states and, consequently, learning from the data they have already used. This creates a rather big impediment when new datasets are introduced.

Fuzzy Logic is a satisfactory solution for a relatively small number of inputs and is able to handle a small to average number of membership functions per input variable. However, when the number of inputs increases, the number of generated Fuzzy rules expands considerably, thus "increasing the computational complexity of the system and decreasing its overall comprehensibility and interpretability" [Jin, 2012].

On the contrary, NNs and ANFIS are faster and more scalable than the Fuzzy Systems. While Fuzzy Systems are mainly tools for knowledge representation, NNs and NFSs retain knowledge and use it for learning purposes of future input features. In NNs there is no need for manual rule inference, parameters are automatically learnt through the provided data and the rule substitutes that are eventually produced by the data norms are "encoded in the network structure" [Siddique and Adeli, 2013]. Despite the rule incorporation, rule extraction is impossible in a NN. "While NNs are good at recognizing patterns, they are not good at explaining how they reach their decisions" [Fullér, 1995]. Moreover, such infrastructures are prone to data over- and under-fitting, thing that can be avoided by an analytic training process and a careful definition of components, such as hidden neurons and layers.

NN operation is almost straightforward, with basic prerequisites, such as ensuring the format of the input and output data matrices. Configuring ANFIS can be a slightly more complicated procedure, because a Takagi-Sugeno system has to be predefined for the learning procedure to begin. This procedure is relatively easily completed in cases of small input spaces, but when the number of fuzzy inputs increases, generation has to occur either from the data grid or via fuzzy clustering.

The current chapter is responsible for adapting the methodology presented in Chapter 2 to the requirements of NNs and ANFIS for the cyberbullying and low level drug dealing use cases, after a background presentation for each system type.

4.1 Neural Networks

NNs were created while aiming to approximate the structure and functionality of actual biological neurons and "mimic the human ability to adapt to changing circumstances and the current environment" [Sivanandam and Deepa, 2006]. In NNs, the human body neuron equivalents, named nodes, act as information processing units. Every node has a theoretically infinite number of inputs and one output. The connections between inputs, nodes and outputs are known by the term "weights" and they represent their strength. During the initialization phase, each input x_n is multiplied by its attributed weight W_n and all the products are summed together, as shown in the following Equation:

$$neuralnet = x_1 \cdot W_1 + x_2 \cdot W_2 + ... + x_n \cdot W_n = \sum_{i=1}^{n}(x_i \cdot W_i) \tag{4.1}$$

Some NN nodes require the presence of an extra parameter value, which allows for function shifting and its role is similar to the parameter b in the linear equation $y = ax + b$. The specific parameter is called bias and the respective Equation 4.1 is transformed in the following manner:

$$neuralnet = \sum_{i=1}^{n}(x_i \cdot W_i) + b \tag{4.2}$$

A schematic representation of a neural node can be found in Figure 4.1. A measure so as to prevent output calculation from falling within non-acceptable ranges is the addition of an activation function f_{act}. The activation function is specifically responsible for strengthening rather weak or small input signals. As a result, the NN output is expressed as $y = f_{act}(neuralnet)$.

The most common activation functions encountered in different NN types are the following:

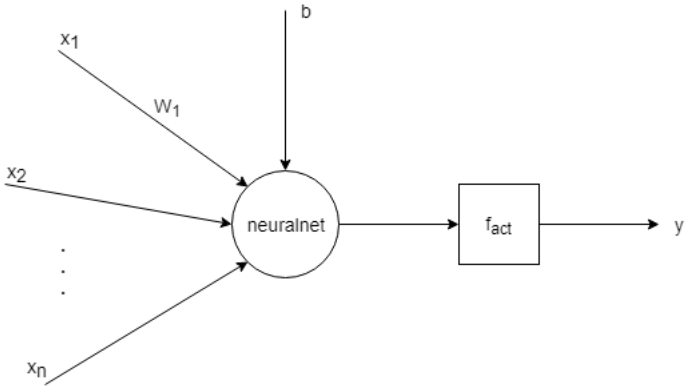

Figure 4.1: Neural node sample, adapted from Siddique and Adeli [Siddique and Adeli, 2013]

Linear

A linear activation function is the product of the output and a constant parameter, i.e.: $y = f_{act}(neuralnet) = C \cdot neuralnet$.

Step

The step activation function receives only two output values, ±1, according to the respective output sign.

$$y = f_{act}(neuralnet) = \begin{cases} +1\,, neuralnet > 0 \\ -1\,, neuralnet < 0 \end{cases} \tag{4.3}$$

Ramp

The ramp activation function is a hybrid linear and step function. Firstly, a lower and an upper output value bounds are set. The function takes the linear format within the upper and lower bounds, whereas it behaves as a step function everywhere else. More precisely, it receives a maximum value for outputs greater than the upper bound and a minimum value for outputs less than the lower bound. The aforementioned relationship is best described in the following Equation.

$$y = f_{act}(neuralnet) = \begin{cases} max & , neuralnet > u.b. \\ C \cdot neuralnet & , l.b. < neuralnet < u.b. \\ min & , neuralnet < l.b. \end{cases} \tag{4.4}$$

Tansigmoid

A tansigmoid function has the shape of the letter S, "is relatively easy to differentiate and can be formed by a variety of mathematical expressions" [Siddique and Adeli, 2013]. One of the most frequently encountered formats is the following:

$$y = f_{act}(neuralnet) = \frac{1 - e^{neuralnet}}{1 + e^{neuralnet}} \tag{4.5}$$

4.1.1 Neural Networks Architecture

A node functionality and its respective activation function are structural elements of various NN architectures. Before proceeding to their analysis, it is wise to present the main components that are omnipresent in every NN architecture. A NN consists of three main parts: the input, the hidden (intermediate or invisible) and the output layers.

Input layer

The input layer interacts with the environment related to the phenomena under investigation. It consists of a theoretically infinite number of nodes, each one of which represents a value, feature or signal related to the problem to be solved. Input values often undergo a normalization procedure, so as to comply with the upper and lower bounds defined by the activation functions.

Hidden layers

The hidden layers are regulating the main computational activity of the NN and create the patterns which constitute the problem solution.

Output layer

This layer consists of the system outputs, which utilize the processing load of the previous layers in order to calculate the final results.

The most common NN architecture types encountered throughout the respective literature are single-layer feedforward NNs, multi-layer feedforward NNs, recurrent NNs and mesh NNs.

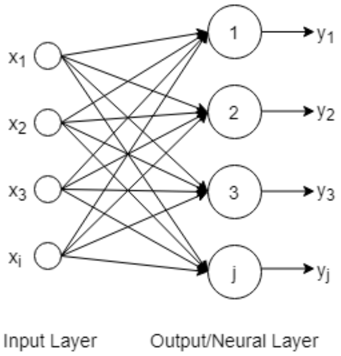

Figure 4.2: Single-layer feedforward NN, adapted from da Silva et al. [da Silva et al., 2017]

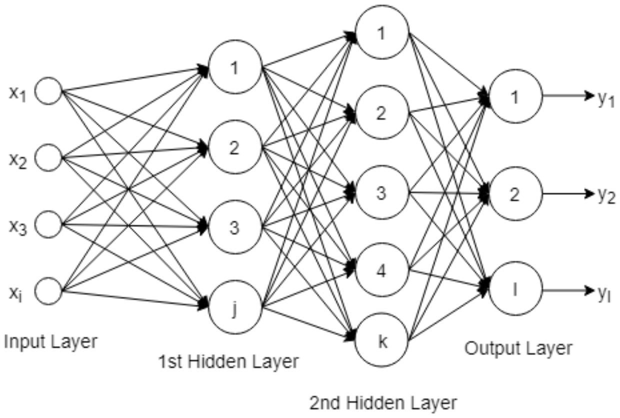

Figure 4.3: Multi-layer feedforward NN, adapted from da Silva et al. [da Silva et al., 2017]

4.1.1.1 Single-layer feedforward Neural Networks

A single-layer feedforward NN comprises an input and an output layer, with the latter also being responsible for any computational process occurring. In the particular NN type, “forward neuron connectivity is the only available option” [Siddique and Adeli, 2013]. Figure 4.2 depicts a NN with i input and j output nodes. As it can easily be inferred by the figure, in such an architecture the number of neurons will always be equal to the number of outputs. “These networks are usually employed in pattern classification and linear filtering problems” [da Silva et al., 2017]. Adaptive Linear Neuron (ADALINE) [Widrow and Hoff, 1960] is a sample case of a single-layer feedforward NN.

4.1.1.2 Multi-layer Feedforward Neural Networks

Similarly to their aforementioned simplified alternative, multi-layer feedforward NNs also use unidirectional neuron connectivity. However, multi-layer feedforward networks consist of one-to-many hidden layers with different number of nodes. They are used for more complex and non-linear problems, such

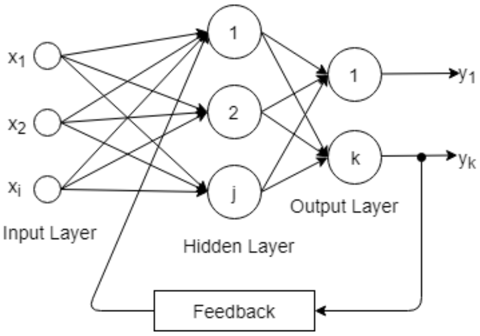

Figure 4.4: Recurrent NN, adapted from da Silva et al. [da Silva et al., 2017]

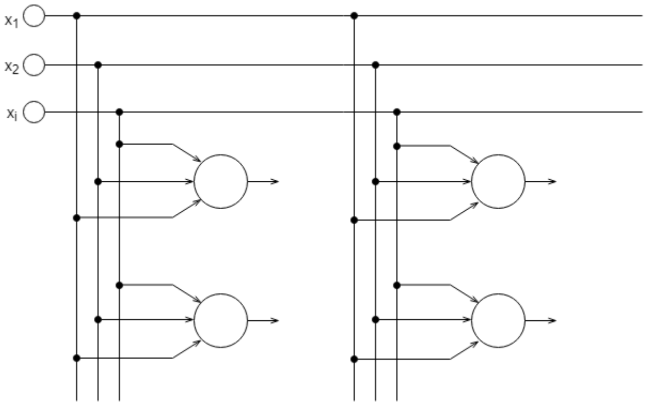

Figure 4.5: Kohonen (mesh) NN, adapted from da Silva et al. [da Silva et al., 2017]

as feature classification and pattern recognition and they are rather scalable. Moreover, the number of hidden layers and nodes depends on the problem complexity and the size of the data sample for examination. Figure 4.3 presents such a NN architecture.

4.1.1.3 Recurrent Neural Networks

Recurrent NNs adopt a different operational model than the previous two categories. The network operates in a controlled but continuous loop manner, where the outputs serve as the new round of inputs for the system, thus enhancing its learning potential. This fact makes the particular architecture ideal for “time series prediction, system identification and optimization, process control, and so forth” [da Silva et al., 2017]. Figure 4.4 represents a Recurrent NN.

4.1.1.4 Mesh Neural Networks

Mesh NNs also have their own unique approach. More precisely, “the spatial localization of the neurons is directly related to the process of adjusting their synaptic weights and thresholds" [da Silva et al., 2017]. A noteworthy example of a Mesh NN architecture is the Kohonen NN [Kohonen, 1982]. Due to the fact that it uses unsupervised learning, a competitive algorithm is applied so as to define the neuron with the best performance, in other words the smaller Euclidean distance between the weight and the node. Figure 4.5 depicts an instance of a Kohonen NN, where all input signals are accessible by every NN node.

4.1.2 Neural Network training

The role of a NN is summarized into apprehending the behavioural fingerprint of a system by establishing relations between its inputs and outputs and by optimizing potential solutions by generalization. In the same line of reasoning, this is achieved by tweaking the weights, biases and node thresholds in a step-alike manner that leads to production of better results by correction of the previous values and continuous learning. The aforementioned procedure is also known as NN training.

The first step of the training process is the dataset division into two subsets. The training subset occupies 60-90% of the initial dataset, whereas the testing dataset occupies 10-40%. Some other approaches adopt the use of an additional validation dataset, which usually occurs by splitting the test data in half. A validation dataset is "an unbiased estimate of the results produced during the training procedure" [Brownlee, 2017]. Extra precautions need to be taken during the creation of all the aforementioned datasets, so as to ensure their statistical integrity [da Silva et al., 2017]. In the current thesis, we use an approach consisting of 70% of the initial data sample for training, 15% for testing and the remaining 15% for validation purposes, whenever necessary.

NN training comprises three main categories of learning rules; supervised, reinforcement and unsupervised learning.

4.1.2.1 Supervised Learning

In supervised learning, the NN is introduced to a model of correct behaviour, represented by the following set of tuples, where x_i are the inputs and t_i are the desired (target) output values. Equation 4.6 presents such a set of tuples. "As the inputs are applied to the network, the network outputs are compared to the targets" [Demuth et al., 2014]. The learning rule is responsible for performing the appropriate parameter adjustments so as to produce outputs that achieve the highest possible level of approximation to the target values. In other words, supervised learning constitutes an error reduction methodology. Some classic examples of supervised learning rules are the "Widrow–Hoff rule, Gradient descent, the Delta rule, the Backpropagation rule, the Cohen–Grossberg learning rule, and the Adaptive conjugate gradient model of Adeli and Hung" [Siddique and Adeli, 2013].

$$\{x_1, t_1\}, \{x_2, t_2\}, ..., \{x_i, t_i\} \tag{4.6}$$

4.1.2.2 Reinforcement Learning

Reinforcement learning is a rule that operates by the supervised learning principles. The difference between the two methods is the fact that instead of using targets, the reinforcement learning rules compute a score as a means of comparison. “The network learning process is usually done by trial and error because the only available response for a given input is whether it was satisfactory or unsatisfactory in comparison to the calculated score” [da Silva et al., 2017]. If the outcome is positive, the NN rewards (reinforces) the specific approach. Reinforcement learning applications are not as widespread as their supervised learning equivalents. However, they target control systems with noteworthy results in agent behaviour replication [Saikia et al., 2011], [Mnih et al., 2015].

4.1.2.3 Unsupervised Learning

In unsupervised learning methods, the NN parameters are only affected by the respective inputs. No target outputs exist. The NN “needs to organize itself when there are existing particularities between the elements that compose the entire sample set, identifying subsets presenting similarities” [da Silva et al., 2017]. As a result, unsupervised learning is more efficiently applied to circumstances where the potential outcome is not known a-priori and most of the conclusions are extracted after observation, taking into consideration that the majority of the existing perceptrons “mainly perform clustering operations” [Demuth et al., 2014]. The next section presents the backpropagation algorithm, that is used for the experimental part of this thesis.

4.1.3 Backpropagation Algorithms

Backpropagation is the most frequently used method for gradient calculation of NN weights. “The computed gradient is employed so as to determine the weights that minimize the error” [Garcia and Zhou, 2010], as it is defined by the Delta rule in Equation 4.7, where t_i are the targets and y_i the actual outputs of the NN.

$$E = \frac{1}{2}(t_i - y_i)^2 = \frac{1}{2}e^2 \tag{4.7}$$

Let the NN in Figure 4.3 be a sample multi-layer perceptron. The first phase of the backpropagation algorithm, i.e. forward propagation, begins when the x_i input values are introduced into the perceptron. Propagation begins with randomly initiated parameters (biases and weights) until the last layer, where the actual outputs y_l are calculated. Afterwards, they are compared to the targets and the error is computed. The second phase, i.e. backward propagation follows the opposite layer-by-layer direction and the NN parameters are adjusted accordingly.

The following subsections describe in detail the different variations of backpropagation algorithms.

4.1.3.1 Levenberg-Marquardt Backpropagation

The Levenberg-Marquardt backpropagation algorithm is a hybrid of the steepest descent method and the Gauss-Newton algorithm. Its main goal is the optimized minimization of the global error function $E(x, w)$ depicted in Equation 4.8, where x and w are the input and weight vectors, m and n are the total numbers of patterns p and outputs y and $e_{p,y}^2$ is the squared value of the error.

$$E(x, w) = \frac{1}{2} \sum_{p=1}^{m} \sum_{y=1}^{n} e_{p,y}^2 = \frac{1}{2} \sum_{p=1}^{m} \sum_{y=1}^{n} (t_{p,y} - o_{p,y})^2 \tag{4.8}$$

The steepest descent method utilizes the first-order derivative of the global error function, so as to determine the minima within the error space, a characteristic that renders it “optimal for areas with complex curvature” [Wilamowski and Yu, 2010]. When the aforementioned curvature allows for quadratic approximation, the respective error minimization algorithm selected is Gauss-Newton. In the Gauss-Newton algorithm, the weights vector is represented as a set of linearly independent gradient functions that are all set to zero for the estimation of the global error minima. Contrary to the steepest descent method, the second-order derivatives of the global error function, known as Hessian matrix H are calculated. In order to avoid complications caused by the calculation complexity of H driven by the second-order derivatives, the Jacobian matrix J is introduced instead, since, for the specific circumstances, H can be approximated as shown in Equation 4.9 [Bui et al., 2012].

$$H \approx J^T J \tag{4.9}$$

The combination coefficient μ, a positive value multiplied by the identity matrix I is added to Equation 4.9, so as to ensure that H remains always invertible. As a result, the Levenberg-Marquardt algorithm uses the following Equation.

$$H \approx J^T J + \mu I \tag{4.10}$$

The weights of a perceptron trained with the Levenberg-Marquardt algorithm are calculated by the following Equation, where w_ℓ and e_ℓ are the weight and error value of the ℓ-th node.

$$w_{\ell+1} = w_\ell - (J_\ell^T J_\ell + \mu_\ell I)^{-1} J_\ell e_\ell \tag{4.11}$$

When μ approaches values close to zero, the Gauss–Newton algorithm is used; when μ obtains a large value, the training algorithm swaps to the steepest descent method.

4.1.3.2 Bayesian Regularization Backpropagation

The Bayesian backpropagation algorithm is claimed to enhance the protection mechanism against perceptrons‘ overfitting and overtraining issues [Ticknor, 2013]. It “combines the conventional sum of the least squares error function with an additional term called regularization” [Bui et al., 2012]. This term, when added to the sum squared error equation, prevents the function from getting trapped into local minima [Burden and Winkler, 2009] and the following cost function $S(w)$ is created, where α and β are the hyperparameters, E_p is the sum of squared errors and E_w “is the penalty term, which penalizes large values of the weights” [Bui et al., 2012], with n being the maximum number of weights.

$$S(w) = \beta E_p + \alpha E_w = \beta \sum_{p=1}^{n} (t_p - o_p)^2 + \alpha \sum_{q=1}^{n} w_i^2 \tag{4.12}$$

The perceptron weights are regarded as random variables within the context of a Bayesian network [Foresee and Hagan, 1997]. As a result, the Bayes‘ theorem can be applied for the presentation of their density function.

$$P(w|X, \alpha, \beta, N) = \frac{P(X|w, \beta, N) P(w|\alpha, N)}{P(X|\alpha, \beta, N)} \tag{4.13}$$

In Equation 4.13, X is the input data vector, w refers to the perceptron weights‘ vector, while N is the perceptron model utilized.

4.1.3.3 BFGS Quasi-Newton Backpropagation

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm belongs to the quasi-Newton family of algorithms, which, contrary to Newton algorithms do not demand the calculation of a Hessian matrix due to potential lack of positive definite results or plain inefficiency [Xia et al., 2010]. They instead use a "symmetric positive definite approximation matrix A_n based on a rank-two correlation method" [Ghosh and Chakraborty, 2012]. The inexact search scheme utilized "improves the computational scheme and allows the algorithm to have a global convergence property" [Fletcher, 1987]. The first step of the configuration includes the definition of a search direction S_n, where g_n is the gradient vector for each iteration.

$$s_n = -A_n^{-1} g_n \tag{4.14}$$

Once the direction is encountered, the search continues alongside so as to discover a step length σ_n that satisfies the following criterion.

$$f(w_n + \sigma_n s_n) = \min_{\sigma \geq 0}(f(w_n + \sigma_n s_n)) \tag{4.15}$$

Under the current circumstances, the weight vector w_n for the following step is formed as shown in Equation 4.16.

$$w_{n+1} = w_n + \sigma_n s_n \tag{4.16}$$

Afterwards, A_n gets updated to A_{n+1} with the rank-two correction provided below, where $\beta_n = w_{n+1} - w_n$ and $\gamma_n = g_{n+1} - g_n$.

$$A_{n+1} = A_n + \frac{\gamma_n \gamma_n^T}{\gamma_n^T \beta_n} - \frac{A_n \beta_n A_n \beta_n^T}{\beta_n^T A_n \beta_n} \tag{4.17}$$

The iteration comes to an end when a provided value λ is marginally greater or equal to the gradient of the objective function.

$$\sqrt{g_n^T g_n} \leq \lambda \tag{4.18}$$

4.1.3.4 One-Step Secant Backpropagation

The One-Step Secant (OSS) backpropagation algorithm was created as a means of "bridging the gap between the quasi-Newton and the Conjugate Gradient families of algorithms" [Kuruvilla and Gunavathi, 2014]. In order to avoid storage issues, it omits storing the entire Hessian matrix and another approach is adopted instead. OSS is actually a memory-less BFGS variation. Instead of requiring an $O(N^2)$[20] amount of calculations order, its storage needs are reduced to $O(N)$. This is achieved "by obtaining the positive definite secant update for the inverse matrix A_{n+1}^{-1}" [Battiti, 1992] from Equation 4.18.

$$\begin{aligned} A_{n+1}^{-1} = A_n^{-1} + \frac{(\beta_n - A_n^{-1}\gamma_n)\beta_n^T + \beta_n(\beta_n - A_n^{-1}\gamma_n^T)}{\gamma_n{}^T\beta_n} \\ - \frac{< \beta_n - A_n^{-1}, \gamma_n > \beta_n^T\beta_n}{(\gamma_n^T\beta_n)^2} \end{aligned} \tag{4.19}$$

Once each iteration ends, it is automatically inferred that the Hessian matrix of the previous step was the identity matrix I [Singh et al., 2006]. If Equation 4.19 is multiplied by the error gradient $g_n = \nabla(E(x,w))$, the next search direction (s_w) will be:

$$\begin{aligned} s_w = -g_n + C_n\beta_n + D_n\gamma_n = -g_n \\ +[(-1 - \frac{\gamma_n^T\gamma_n}{\beta_n^T\gamma_n})\frac{\beta_n^T g_n}{\beta_n^T\gamma_n} + \frac{\gamma_n^T g_n}{\beta_n^T\gamma_n}]\beta_n + (\frac{\beta_n^T g_n}{\beta_n^T\gamma_n})\gamma_n \end{aligned} \tag{4.20}$$

4.1.3.5 Resilient Backpropagation

The main scope of Resilient backpropagation is to diminish the "adverse effects magnitudes of partial derivatives" [Chabaa et al., 2010] may cause to the weight update process, and consequently to the minimization of the error function. Thus, only the gradient signs are taken into consideration. The process is depicted in Equation 4.21.

$$w_n - w_{n-1} = -sign(g_{n-1})\Delta_n \tag{4.21}$$

The same initial Δ_0 is assigned to every update value. If the product of the current and the previous step is a positive number, i.e. if the vectors have the same direction, then a value η^+, greater than 1, is multiplied by the update value. Otherwise, the product of η^-, a negative value less than 1, and the update value is calculated. The aforementioned relationship is depicted in Equation 4.22.

$$\Delta_n = \begin{cases} \eta^{+}\Delta n - 1\,, g_{n-1}g_n > 0 \\ \eta^{-}\Delta n - 1\,, g_{n-1}g_n < 0 \end{cases} \tag{4.22}$$

4.1.3.6 Conjugate Gradient Backpropagation Family

Contrary to the steepest descent backpropagation algorithms, which use the negative gradient direction so as to achieve a faster reduction of the error function rates, the conjugate gradient family of backpropagation algorithms shares another common principle. Their basic aim is the production of more rapid convergence rates, so search is performed in the span of conjugate directions [Shaharin et al., 2014] and "the step size is adjusted at each iteration" [Kuruvilla and Gunavathi, 2014].

The very first iteration in all the members of the Conjugate Gradient family of algorithms sets the search direction towards the negative gradient. Equation 4.23 shows the particular relationship, with s_0 indicating the search gradient and g_0 the initial gradient.

$$s_0 = -g_0 \tag{4.23}$$

Afterwards, linear search is employed so as to encounter the most appropriate moving distance. Equation 4.24 shows the status of the next weight w_{n+1}, which is the sum of the actual weight w_n and the product of the learning rate λ and the current search gradient s_n.

$$w_{n+1} = w_n + \lambda_n s_n \tag{4.24}$$

A prerequisite for the selection of the next search direction is to remain conjugate to the previous ones. The search direction is then defined anew as the combination of "the new steepest descent direction with the previous search direction" [Kuruvilla and Gunavathi, 2014]. More details are provided in Equation 4.25, where s_{n-1} is the last search direction and β_n is a constant value that may vary from algorithm to algorithm.

$$s_n = -g_n + \beta_n s_{n-1} \tag{4.25}$$

Fletcher-Reeves Updates

For the Fletcher-Reeves updates variation, the search direction is defined by Equation 4.25 and the aforementioned constant β_n is defined as ratio of the squared norm of the current gradient and the squared norm of the last one.

$$\beta_n = \frac{||g_n||^2}{||g_{n-1}||^2} \tag{4.26}$$

Polak-Ribière Updates

Similarly to the Fletcher Reeves variation, the Polak-Ribière updates use Equation 4.25 for the calculation of the search direction. However, β_n is defined as the division of "the inner product of the previous change in the gradient with the current gradient divided by the square of the previous gradient" [Kuruvilla and Gunavathi, 2014].

$$\beta_n = \frac{\Delta g_n^T g_n}{g_{n-1}^T g_{n-1}} \tag{4.27}$$

This method, as a four-vector algorithm, requires a bit higher amount of storage resources that the three-vector Fletcher-Reeves equivalent.

Powell-Beale Restarts

All members of the Conjugate Gradient family require a certain, periodic amount of resets to the original negative gradient value. Each restart takes place as soon as the number of iterations performed reaches the total number of the perceptron's biases and weights. However, restarts are not exclusively obligatory, but can rather occur at any given moment during the training phase, so as to improve its overall performance. The Powell-Beale restarts take place whenever the absolute value of the product $g_n^T g_n$ is greater than or equal of 0.2 times the squared norm of the current gradient. In terms of storage, the six-vector Powell-Beale restarts require more space than the two aforementioned variations.

$$|g_{n-1}^T g_n| \geq 0.2||g_n||^2 \tag{4.28}$$

Scaled Conjugate Gradient

Contrary to the rest Conjugate Gradient algorithms, the Scaled Conjugate Gradient variation avoids performing searches per-line iteration and "uses a step-sized scaling mechanism instead" [Shaharin et al., 2014]. Due to computational complexity, the Hessian matrix H is approximated as shown in Equation 4.29, where w_n is the n-th weight, p_n is the n-th search direction, E‘ is the error gradient and σ_n and λ_n are scaling factors predefined by the user [Baghirli, 2015].

$$H_n = \frac{E‘(w_n + \sigma_n p_n) - E‘(w_n)}{\sigma_n} + \lambda_n p_n, \\ \sigma \in (0, 10^{-4}), \lambda \in (0, 10^{-6}) \tag{4.29}$$

Additionally, the constant β_n is defined as follows:

$$\beta_n = \frac{|g_{n+1}|^2 - g_{n+1}^T g_n}{g_n^T g_n} \tag{4.30}$$

4.1.3.7 Gradient Descent Backpropagation Family

The Gradient Descent family of algorithms contains some of the simplest backpropagation implementations. In this thesis, the plain Gradient Descent and its variations (Gradient Descent with Momentum and Gradient Descent with Momentum and Adaptive Learning Rate) are examined.

Gradient Descent

Gradient Descent constitutes one of the plainest forms of error minimization techniques. The weight update is introduced as the product of the learning rate λ and the negative error gradient.

$$\Delta w(n) = -\lambda \nabla E(w) \tag{4.31}$$

Deciding on an appropriate learning rate is a rather difficult task and depends highly on the shape of the error function. Moreover, too high or too low values may lead to poor performance. One of the most known issues that the Gradient Descent algorithm faces is the local minima trap, that does not allow for further (global) minimization of the error function.

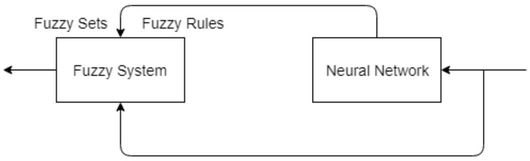

Figure 4.6: Cooperative NFS, adapted from Vieira et al. [Vieira et al., 2004]

Gradient Descent with Momentum

A simple variation of the Gradient Descent algorithm is the addition of a momentum term. In other words, “the parameter μ scales the influence of the previous weight-step on the current one”[Riedmiller, 1994].

$$\Delta w(n) = -\lambda \nabla E(w) + \mu \Delta w(n-1) \tag{4.32}$$

Gradient Descent with Momentum and Adaptive Learning Rate

Momentum by itself is not enough to avoid complications deriving from rather poor choice of learning rate values. For this purpose, the algorithm is enhanced by the adaptive learning rate, which “attempts to maintain the learning step size as large as possible while, keeping learning stable” [Mohanty et al., 2010].

$$\Delta w(n) = -\lambda \mu \nabla E(w) + \mu \Delta w(n-1) \tag{4.33}$$

The following section presents the basic concepts of Neuro-Fuzzy Systems (NFSs) and elaborates the characteristics of the Adaptive Neuro-Fuzzy Inference System (ANFIS), which is going to be used in this thesis.

4.2 Neuro-Fuzzy Systems

NFSs are a hybrid concatenation of Fuzzy Systems and NNs, so as to profit from all the characteristics of both techniques and avoid the impediments each method poses separately. A NFS is defined as “a NN that uses a fuzzy approach to enhance learning capabilities and improve performance” [Siddique and Adeli, 2013]. There are three different types of NFSs encountered in the respective literature, namely cooperative, concurrent and hybrid NFSs.

In a cooperative NFS, the existing Fuzzy System and NN are two independent entities. Each one contributes in the configuration of the other‘s parameters, according to the initial setup provided. This divides the cooperative NFSs into two different categories: the Fuzzy-NN cooperative system, where all the

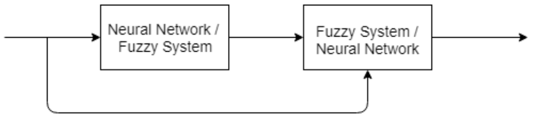

Figure 4.7: Concurrent NFS, adapted from Vieira et al. [Vieira et al., 2004]

parameters that constitute the problem base are provided by the fuzzy system settings and the NN performs the learning and configuration procedure and the NN-Fuzzy cooperative system, where the fuzzy parameters are defined by the NN operation. Simultaneous and bidirectional flow is feasible, but its computational efficiency is debatable [Vieira et al., 2004]. Figure 4.6 represents a cooperative NFS.

In a concurrent NFS, both the NN and the Fuzzy System operate in a parallel independent manner, without influencing eachother and they both contribute to the optimization of the final problem outcome. An instance of a concurrent NFS is shown in Figure 4.7. Lastly, in a hybrid NFS the Fuzzy System is incorporated in a NN architecture. For simplification purposes, the fuzzy rule base is substituted by a NN and the fuzzy parameters are inferred via the learning procedure of the network.

NFSs presented substantial evolution during the late 1990s. Some noteworthy examples include Neuro-Fuzzy Control (NEFCON) [Nauck and Kruse, 1994], Fuzzy Adaptive Learning Control Network (FALCON) [Lin and Lee, 1991], Generalized Approximate Reasoning-based Intelligence Control (GARIC) [Berenji and Khedkar, 1992], Adaptive Network-based Fuzzy Inference System (ANFIS) [Jang, 1993], Fuzzy Inference and Neural Network in Fuzzy Inference Software (FINEST) [Tano et al., 1996], Fuzzy Net (FUN) [Sulzberger et al., 1993], Self Constructing Neural Fuzzy Inference Network (SONFIN) [Juang and Lin, 1998], Fuzzy Neural Network (NFN) [Figueiredo and Gomide, 1999] and Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) [Kasabov and Song, 2002]. Despite the diversity the discipline showed until the early 2000s, current research is almost entirely focused on ANFIS and its applications [Kar et al., 2014], new models are produced in a lower pace and they are usually modifications of their predecessors [Viharos and Kis, 2015].

It is a certainty that there is a plethora of already developed and evaluated NFSs. However, their availability and documentation for academic use are rather limited. In this perspective, ANFIS was selected as the more appropriate tool for the current research. An introduction to its operation is provided in the following section.

4.2.1 ANFIS

ANFIS was first introduced by Jang [Jang, 1993] and combines the precision of Fuzzy Logic calculations with the adaptability of NNs. It eradicates the learning incapability of Fuzzy Systems, while it simultaneously makes use of

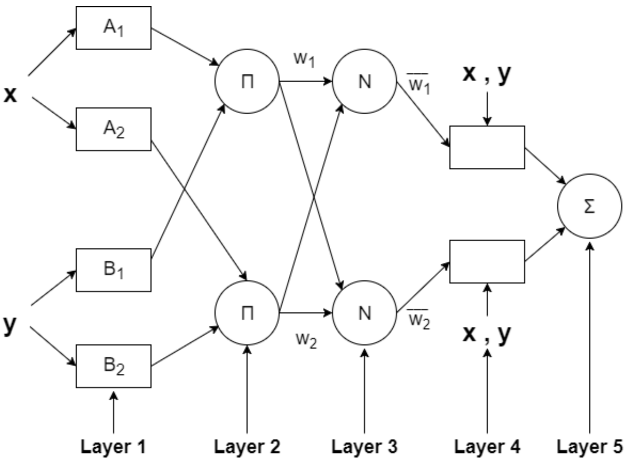

Figure 4.8: ANFIS, adapted from Jang [Jang, 1993] and Suparta and Alhasa [Suparta and Alhasa, 2016]

the NN learning methods so as to efficiently tune fuzzy parameters, such as membership functions and their positioning. The learning algorithm present in ANFIS is inspired by the structure of a Takagi-Sugeno Fuzzy System. A plain form, comprising two inputs, one output and a base of two rules is adopted for demonstration purposes.

$$\textit{Rule 1: if } x = A_1 \textit{ and } y = B_1, \textit{ then } f_1 = k_1x + l_1y + m_1$$
$$\textit{Rule 2: if } x = A_2 \textit{ and } y = B_2, \textit{ then } f_2 = k_2x + l_2y + m_2$$

While x and y are the inputs, A_1, A_2, B_1 and B_2 are the membership functions corresponding to parts of the linguistic variables. The values k_1, k_2, l_1, l_2, m_1 and m_2 constitute linear parameters.

A simplified representation of ANFIS is provided in Figure 4.8. The first layer comprises the system inputs. Contrary to the aforementioned perceptrons, each input does not correspond to the full range of a variable, but rather to separate partial fuzzy membership of it. From the previous claim, it can easily be inferred that the ANFIS input space is more complicated and fragmented than the one belonging to a plain or pattern recognition perceptron. The output produced by the first layer is provided by the following Equation, where μ is the membership function per input partition, usually Gaussian or Bell and $i \in Z_+^*$.

$$_{1,i} = \begin{cases} \mu_{A_i}(x) \\ \mu_{B_i}(y) \end{cases} \tag{4.34}$$

The second layer consists of fixed Π nodes and corresponds to rule formulation by combining the appropriate membership values. Their output is calculated after the addition of a "T-norm operator to the existing membership value and represents the firing strength of each rule" [Suparta and Alhasa, 2016]. Its format is shown in Equation 4.35.

$$_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \tag{4.35}$$

The third layer also comprises non-adaptive N nodes and performs normalization of each rule‘s firing strength. This is achieved by dividing the current strength of the rule by the total strength of every rule encountered in the system.

$$_{3,i} = \overline{w_i} = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{4.36}$$

The fourth layer produces node functions as outputs, the structure of which is provided below. The set of $\{k_i, l_i, m_i\}$ is known as a consequent parameters set.

$$_{4,i} = \overline{w_i} f_i = \overline{w_i}(k_i x + l_i y + m_i) \tag{4.37}$$

Finally, the fifth layer is responsible for the global output calculation. The global output is defined as "the summation of all incoming signals" [Jang, 1993].

$$_{5,i} = \sum_{i=1}^{n} \overline{w_i} f_i \tag{4.38}$$

In order to avoid the local minima trap issue deriving from the use of traditional backpropagation algorithms, such as Gradient Descent, a hybrid version, consisting of two opposite direction paths, was proposed. During the forward path, signals reach the fourth layer and a Least Square Estimate (LSE) algorithm is employed so as to determine the set of consequent parameters. Once this step is complete, the new data are deployed as inputs, and then the respective outputs are calculated and compared to the target outputs. "The consequent parameters remain in a steady state for the backward path" [Suparta and Alhasa, 2016], as a reference point. The error resulting from the aforementioned comparison is forwarded anew to the first layer and Gradient Descent or other backpropagation algorithms are used for further optimization and final convergence.

Before proceeding to the utilization of the three different perceptrons, it is advisable to normalize the respective inputs and outputs. Normalization is the procedure of "rescaling the input and output variables independently by the minimum and range of the vector, to make all the elements lie in the set of [0,1]" [Iglewicz, 1983]. This procedure can either take place manually, or automatically by the features of the Matlab Neural Network Toolbox.

After the presentation of the intelligent computation methods of this chapter, the following section elaborates the methodology adaptation to them.

4.3 Methodology

This section presents the adaptation of the methodology presented in Chapter 2 to the needs of both NNs and ANFIS. Most of the methodology steps corresponding to the data preparation are common for all the techniques and only the application and evaluation procedures show some differences, which are presented analytically for each one of them.

4.3.1 Use Case Definition

The discipline of Mobile Forensic Data Analysis (MFDA) handles acquired artifacts from devices that are either compromised by malicious entities (malware propagation, bitcoin miners, botnet zombies) or serve as means to facilitate the conduction of a crime [Barmpatsalou et al., 2013]. The current thesis focuses on the latter category, which is the one more strongly correlated to human behaviour. More precisely, it aims to pinpoint different criminal activities to specific metadata patterns. We selected two offender types for the current examination. The selection was performed according to the public availability of information concerning the offenders' involvement with mobile devices. We queried content related to different criminal digital profiles in Psychology, Sociology, Law and IT journals and the cases that allowed for a higher level of analysis due to their availability and abundance of information were cyberbullying and low-level drug dealing.

In order to proceed to the detection and correlation procedure, each offender's Modus Operandi (MO) has to be defined. Once the characteristics are outlined, rules related to the suspiciousness of data patterns can be inferred and the ground truth can be generated. Law enforcement has long held to the belief that understanding the methods and techniques criminals use to commit crime is the best way to "investigate, identify, and ultimately apprehend them" [Turvey, 2011]. The following paragraphs analyze the MO of the two aforementioned offender types.

4.3.1.1 Cyberbullying

Bullying by mobile devices seems to be a growing trend and "was perceived to have a rather negative impact" [Smith, 2008]. One of the main characteristics of cyberbullies is the very frequent use of their mobile devices, especially for texting via the native or other downloaded applications [J. T. Fish, L. S. Miller, M. C.

Braswell, and E. W. Wallace Jr., 2014], [Lievens, 2014], [Roberto et al., 2014], [Goodboy and Martin, 2015]. However, this fact by itself cannot be considered a framing factor, since teenagers are classified as heavy mobile device users. Cyberbullies tend to send massive text messages [Smith, 2008], [D. T. Sacco and Tallon, 2010], [Roberto et al., 2014] and they prefer to "attack" their victims after school, especially at night, when their activity is usually not monitored by parental controls [Gorzig and Olafsson, 2013], [Openet, 2013], [J. T. Fish, L. S. Miller, M. C. Braswell, and E. W. Wallace Jr., 2014]. The messages they send are not long; however, they just tend to be annoying by sending small to medium-sized, but insulting texts [Openet, 2013], [Roberto et al., 2014]. Moreover, they also perform many missed or low duration calls to the victims, in order to bother them more or to even provoke them to reply in case they decided to ignore them [Smith, 2008], [Roberto et al., 2014].

A cyberbully's MO shows relatively intense device usage and thus facilitates the inference of a digital fingerprint. The next paragraph highlights a low-level drug dealer's MO.

4.3.1.2 Low Level Drug Dealing

Low level drug dealing targets dealers of small quantities, who interact more with potential buyers and less with cartel leaders or other providers. As a result, the majority of their call and message exchanges takes place among entities within the same country [Natarajan, 2006], [McEwen, 2011], [Fleetwood, 2014]. Dealers prefer using mobile devices because they prevent them from increased physical interaction with the clients, which increases the probabilities of being arrested [May and Hough, 2004], [Casey, 2011]. Drug dealers are highly active in terms of message exchange [Natarajan, 2006], [Edwards, 2013], [Fleetwood, 2014] and call performance [McEwen, 2011], [Edwards, 2013], [Fleetwood, 2014]. They also interact frequently with specific people, their clientèle, mainly during evenings and nights. Their calls have small duration and they are usually the ones performing than receiving them, based on their convenience. The text messages they send have medium to relatively long length [Natarajan, 2006], [Barmpatsalou et al., 2017] and contain information about the products they are selling, often mixed with irrelevant phrases.

4.3.2 Datasets

As described analytically in Section 3.3, where the data source issue was discussed, the CDA dataset served as the main investigation material for the current research. We used the tuple encountered in Equation 3.16 as a template for a more complex input setup. The is split in such a way that each column belongs to a unique attribute.

Similarly to the data in Chapter 3, their format is not in the appropriate state to be properly interpreted by a NN or ANFIS. This can be achieved by applying a pre-processing procedure, which is described analytically in the following subsection.

4.3.3 Pre-processing

Pre-processing is related to any data modification that can facilitate their interpretation by the NN perceptrons and ANFIS. Continuous numeric variables do not need further alteration. Linguistic variables that describe different states or numeric variables that denote time frames are translated into numeric discrete values, following the categorization pattern adopted by the authors in a previous paper [Barmpatsalou et al., 2018b]. Lastly, linguistic or date-related variables that are not useful for the investigation in their current formatting are transformed into a summation of their appearance instances. The initial format for both calls and SMS data types used in this research can be found below and a more detailed explanation follows in the following paragraphs. Algorithms 4.1 and 4.2 depict the pre-processing procedure.

$$\texttt{Call}(\texttt{type}, \texttt{timestamp}, \texttt{name}, \texttt{location}, \texttt{number\_type}, \texttt{duration}) \tag{4.39}$$

$$\texttt{SMS}(\texttt{type}, \texttt{timestamp}, \texttt{name}, \texttt{location}, \texttt{number\_type}, \texttt{length}) \tag{4.40}$$

4.3.4 Common Attributes

Both the calls and the SMS datasets have some equal attributes, the pre-processing procedure of which is going to be explained in a common space.

Name

The *Name* attribute corresponds to the name or the phone number of the individual with whom the owner of the device interacted. All the names and numbers in the CDA dataset are anonymized and thus, no sensitive information can be extracted from their raw format. However, the instances of each number lead to the creation of *Appearance Frequency*, a variable concerning the amount of total owner interaction with various other entities by calls or text messages.

Timestamp

Timestamp is a unified string, comprising the date and the time a call was performed or an SMS was sent or received. This string is later split into the *Date* and *Time* attributes. Despite the fact that the date itself is a useful observation in terms of a digital investigation, it does not provide useful insights

Table 4.1: Time quantification

Time of the day	**Value**
Morning (05:01-12:00)	0
Afternoon (12:01-17:00)	1
Evening (17:01-22:00)	2
Night (22:01-05:00)	3

for the scope of the current research. Thus, it is converted to the *Daily Frequency* variable, which is the amount of interactions a user had within a 24-hour period. The *Time* variable is converted to four discrete categories, according to Table 4.1.

Location

The *Location* attribute is represented by linguistic terms in the CDA dataset and refers to whether the phone number of an entity that interacted with the device owner is foreign, local and unknown or undefined, due to parsing errors. The generated *Country code* variable has three discrete values that are presented in Table 4.2.

Number type

Similarly to the *Location* attribute, *Number type* consists of strings that describe if the number the user is interacting is mobile, unknown or a fixed line. The generated *Mobility* variable is also present in Table 4.2.

Apart from the common data attributes, there are also two more data categories that correspond exclusively to the calls and SMS types and are described in the following paragraphs.

4.3.5 Call-exclusive Attributes

The calls data type comprises two attributes that are unique and create two different variables.

Type

The call *Type* is a binary variable and receives the value *0* for outgoing and *1* for incoming calls.

Table 4.2: Country code and mobility quantification

Location	**Number Type**	**Value**
Foreign	Fixed Line	0
Unknown/Undefined	Unknown/Undefined	1
Local	Mobile	2

Duration

Duration is also a call-specific continuous variable, which receives positive integer values in seconds. For the missed calls, the value *–1* is assigned. Zero could also be an assigned value for a missed call, but after a careful observation of the original dataset, there were some incoming and outgoing calls with very small duration that received the same value.

4.3.6 SMS-exclusive Attributes

Similarly to the calls, there are also two variables dedicated to the SMS texts.

Type

The SMS *Type* is a binary variable and receives the value *0* for sent and *1* for received SMS messages.

Length

The last SMS-specific attribute, *Length*, is a continuous variable, receives positive integer values and corresponds to the total number of characters in a text message.

In the end of the pre–processing procedure, both calls and SMS data types consist of seven variables and are fully quantified. Thus, they are ready to be used as inputs for the phases of the ground truth generation and the NN and ANFIS training and testing.

4.3.7 Ground Truth Generation

In an earlier paper [Barmpatsalou et al., 2018b], the authors introduced an alternative representation of the output suspiciousness. Instead of using the classic binary format (0: not suspicious - 1: suspicious), the output is a fuzzy variable, receiving values within the [0,1] interval. Values closer to zero are considered innocent, whereas values closer to one are regarded as more

Algorithm 4.1. Calls preprocessing

```
procedure Pre-Processing
    function splitAttributes($RawDataset$) return $SplittedDataset[number\_of\_attributes]$
    end function
    function createCallType($SplittedDataset[type]$)
        $t \leftarrow list()$
        for each line in $SplittedDataset[type]$ do
            if $type = incoming$ then
                $t \leftarrow 1$
            else if $type = outgoing$ then
                $t \leftarrow 0$
            end if
        end forreturn $t$
    end function
    function createCallTime($SplittedDataset[time]$)
        $tm \leftarrow list()$
        for each line in $SplittedDataset[time]$ do
            if $time \leq 12{:}00$ and $time > 05{:}00$ then
                $tm \leftarrow 0$
            else if $time \leq 17{:}00$ and $time > 12{:}00$ then
                $tm \leftarrow 1$
            else if $time \leq 22{:}00$ and $time > 17{:}00$ then
                $tm \leftarrow 2$
            else if $time \leq 05{:}00$ or $time > 22{:}00$ then
                $tm \leftarrow 3$
            end if
        end forreturn $tm$
    end function
    function createCallDate($SplittedDataset[date]$)
        $d \leftarrow list()$
        for each line in $SplittedDataset[date]$ do
            $d \leftarrow count(date)$
        end forreturn $d$
    end function
    function createCallAf($SplittedDataset[name]$)
        $a \leftarrow list()$
        for each line in $SplittedDataset[name]$ do
            $a \leftarrow count(name)$
        end forreturn $a$
    end function
    function createCallCnt($SplittedDataset[place]$)
        $cc \leftarrow list()$
        for each line in $SplittedDataset[place]$ do
            if $place = local$ then
                $cc \leftarrow 2$
            else if $place = unknown$ then
                $cc \leftarrow 1$
            else if $place = foreign$ then
                $cc \leftarrow 0$
            end if
        end forreturn $cc$
    end function
    function createCallMob($SplittedDataset[mobility]$)
        $mob \leftarrow list()$
        for each line in $SplittedDataset[mobility]$ do
            if $mobility = mobile$ then
                $mob \leftarrow 2$
            else if $place = unknown$ then
                $mob \leftarrow 1$
            else if $place = fixed\ line$ then
                $mob \leftarrow 0$
            end if
        end forreturn $mob$
    end function
    function createCallDr($SplittedDataset[duration]$)
        $dr \leftarrow list()$
        for each line in $SplittedDataset[duration]$ do
            $dr \leftarrow duration$
        end forreturn $dr$
    end function
    function createNNInp($t, tm, d, a, cc, mob, dr$)
        join(t, tm, d, a, cc, mob, dr) return $SMSInputs$
    end function
end procedure
```

Algorithm 4.2. SMS preprocessing

```
procedure Pre-Processing
    function splitAttributes($RawDataset$) return $SplittedDataset[number\_of\_attributes]$
    end function
    function createSMSType($SplittedDataset[type]$)
        $t \leftarrow list()$
        for each line in $SplittedDataset[type]$ do
            if $type = received$ then
                $t \leftarrow 1$
            else if $type = sent$ then
                $t \leftarrow 0$
            end if
        end forreturn $t$
    end function
    function createSMSTime($SplittedDataset[time]$)
        $tm \leftarrow list()$
        for each line in $SplittedDataset[time]$ do
            if $time \leq 12{:}00$ and $time > 05{:}00$ then
                $tm \leftarrow 0$
            else if $time \leq 17{:}00$ and $time > 12{:}00$ then
                $tm \leftarrow 1$
            else if $time \leq 22{:}00$ and $time > 17{:}00$ then
                $tm \leftarrow 2$
            else if $time \leq 05{:}00$ or $time > 22{:}00$ then
                $tm \leftarrow 3$
            end if
        end forreturn $tm$
    end function
    function createSMSDate($SplittedDataset[date]$)
        $d \leftarrow list()$
        for each line in $SplittedDataset[date]$ do
            $d \leftarrow count(date)$
        end forreturn $d$
    end function
    function createSMSAf($SplittedDataset[name]$)
        $a \leftarrow list()$
        for each line in $SplittedDataset[name]$ do
            $a \leftarrow count(name)$
        end forreturn $a$
    end function
    function createSMSCnt($SplittedDataset[place]$)
        $cc \leftarrow list()$
        for each line in $SplittedDataset[place]$ do
            if $place = local$ then
                $cc \leftarrow 2$
            else if $place = unknown$ then
                $cc \leftarrow 1$
            else if $place = foreign$ then
                $cc \leftarrow 0$
            end if
        end forreturn $cc$
    end function
    function createSMSMob($SplittedDataset[mobility]$)
        $mob \leftarrow list()$
        for each line in $SplittedDataset[mobility]$ do
            if $mobility = mobile$ then
                $mob \leftarrow 2$
            else if $place = unknown$ then
                $mob \leftarrow 1$
            else if $place = fixed\ line$ then
                $mob \leftarrow 0$
            end if
        end forreturn $mob$
    end function
    function createSMSLn($SplittedDataset[length]$)
        $len \leftarrow list()$
        for each line in $SplittedDataset[length]$ do
            $len \leftarrow length$
        end forreturn $len$
    end function
    function createNNInp($t, tm, d, a, cc, mob, len$)
        join(t, tm, d, a, cc, mob, len) return $SMSInputs$
    end function
end procedure
```

Table 4.3: Fuzzy Suspiciousness Values, adapted from Barmpatsalou et al. [Barmpatsalou et al., 2018b]

Value	**Suspiciousness Level**
0.15	Very Low
0.25	Low
0.5	Medium
0.75	High
1	Very High

Table 4.4: Output Transformations for the Pattern Recognition Perceptron

Value	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**
0.15	1	0	0	0	0
0.25	0	1	0	0	0
0.5	0	0	1	0	0
0.75	0	0	0	1	0
1	0	0	0	0	1

suspicious. Despite the fact that the output can receive any number within the aforementioned interval, five representative values (0.15, 0.25, 0.5, 0.75, 1) were indicated as thresholds for each suspiciousness category. Table 4.3 demonstrates the assignment and the respective linguistic values.

This approach is also adopted in the current thesis and is the basis of the ground truth generation process. Tuple combinations result in one out of the five aforementioned values. However, from the three NN and NFS methods used, only the plain backpropagation perceptron and ANFIS can make proper use of this method. The pattern recognition backpropagation perceptron requires additional output editing, because its format is based on binary states. In this perspective, five outputs are generated instead of one and one of them receives 1 as a value, whereas the rest of them remain 0s. Table 4.4 shows the output transformation for the pattern recognition backpropagation perceptron. In other words, the ground truth output template for the pattern recognition backpropagation perceptron is a 5x5 square diagonal matrix.

Subsection 4.3.1 provides a qualitative overview of the device usage MO for cyberbullying and low-level drug dealing. This information is rather useful for some first degree inferencing, but it cannot be precise enough without the appropriate numerical boundaries. These thresholds can be calculated after taking into consideration the CDA dataset from Subsection 4.3.2, which includes mobile device usage for period of six months. This way, it is easier to define which variable ranges are considered high, medium or low. Each variable present in Equations 4.39 and 4.40 of Subsection 4.3.3 receives a specific value or interval of values and their combination can be translated into a statement, which is then assigned to a degree of suspiciousness. For example, a highly suspicious call for cyberbullying (Suspiciousness == 1) is performed at night, is missed or has a small duration, the bully‘s appearance frequency is relatively high, is performed by a mobile device and belongs to a local number. On the contrary, an innocent

Table 4.5: CDA GT Pattern Distribution

Use Case Cat.	**Cyberbullying Calls**	**Cyberbullying SMS**	**Drug Dealing Calls**	**Drug Dealing SMS**
0.15	2,268	4,701	1,846	7,010
0.25	156	3,532	623	4,029
0.5	59	3,374	18	351
0.75	61	80	17	74
1	55	302	95	525
Total	2,591	11,989	2,591	11,989

call (Suspiciousness == 0.15) is performed in the morning, has a very high or a very low duration, belongs to a fixed local line or a mobile line abroad.

Two main challenges were encountered during the ground truth generation phase. The first challenge was related to the manual labeling of the results, which was a rather time-consuming procedure, but it ensured their originality. However, in a future phase, this particular procedure can be replaced by a similar ground truth generation algorithm. The second challenge concerned the lack of suspicious patterns. During the manual labeling, there were no patterns that were classified as 1, i.e. the top suspiciousness scale. As a result, we had to generate a random number of suspicious patterns per dataset, based on the characteristics that classified them into the specific category. Once the ground truth generation phase is complete, the preparation phase is concluded as well and the data are ready to be processed by the NN perceptrons and ANFIS.

4.4 Neural Networks and ANFIS

Three different neural and neuro-fuzzy network types, a plain backpropagation perceptron, a back-propagation pattern recognition perceptron and ANFIS are trained and tested. For every use case, seventy percent of the calls and SMS datasets is used for training, whereas fifteen percent is used for testing and the remaining fifteen for validation.

4.4.1 Plain and Pattern Recognition Backpropagation Perceptron Configuration

The follow-up procedure after the dataset splitting is equal for the plain and pattern recognition back-propagation perceptrons and different for ANFIS, that is rather straightforward, guided by the Matlab interface. The respective configuration settings will be presented in the following subsections.

The plain and pattern recognition perceptrons have a similar architecture. They consist of three layers, namely input, hidden and output. The input layer comprises seven inputs, as many as the input variables, whereas the output layer

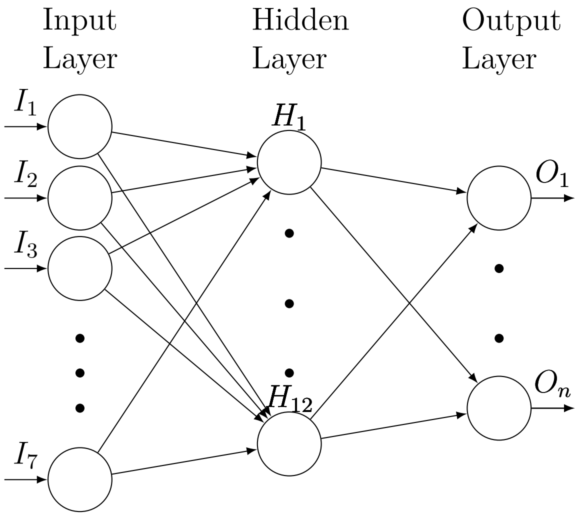

Figure 4.9: Generic plain and pattern recognition perceptron architecture

consists of one output for the plain backpropagation perceptron and five for the pattern recognition backpropagation. The architecture is depicted in Figure 4.9. The decision-making procedure for the hidden layer is more complicated and is analytically discussed in the following paragraphs.

A highly-disputed claim that applies to the cases of the plain and pattern recognition backpropagation perceptrons is the number of hidden layers and nodes that are going to be used. Concerning the hidden layers issue, more than one layers are used in high complexity problems with a big number of inputs, such as image recognition or in case the process being modeled is separable into multiple stages [Frontline Solvers, 2017], whereas problems with a lower number of inputs, such as the one the current thesis is trying to address perform equivalently well with one hidden layer. On the contrary, there are many different arguments related to the number of hidden nodes scattered along the corresponding literature.

One of the approaches by Frontline Solvers [Frontline Solvers, 2017] indicates that there should be an upper bound N_{max} to the number of hidden neurons per layer, which is given by the ratio of the total number of instances in the training dataset N_s, divided by the sum of the number of inputs N_i and outputs N_o, multiplied by an arbitrary scaling factor α that receives values from 2 to 10. More details are provided in Equation 4.41.

$$N_{max} = \frac{N_s}{\alpha * (N_o + N_i)} \tag{4.41}$$

Other bibliographical sources are more precise in terms of defining the exact number of hidden nodes in a perceptron‘s hidden layer, but the diversity of the approaches is relatively big. Some of the referenced claims, broadly known as rule-of-thumb methods can be found below:

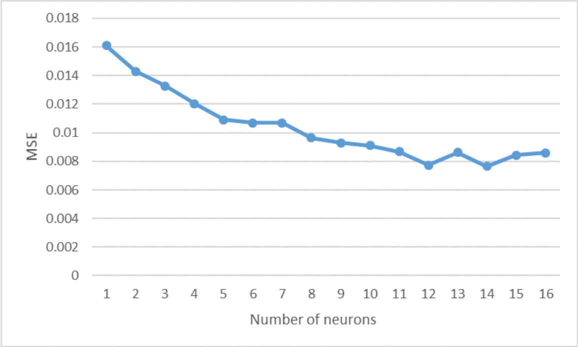

Figure 4.10: Average MSE per number of neurons - BP calls

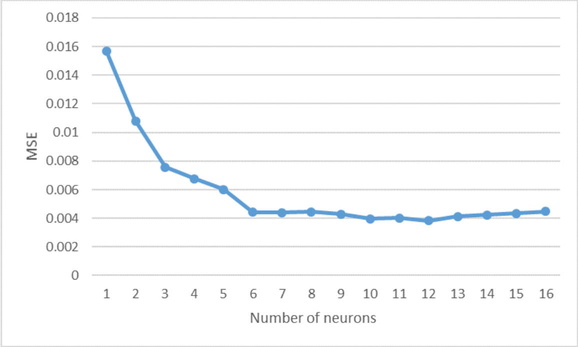

Figure 4.11: Average MSE per number of neurons - BP SMS

- “The number of hidden neurons should be between the size of the input layer and the size of the output layer” [Heaton, 2008].
- “The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer” [Panchal et al., 2011].
- “The number of hidden neurons should be less than twice the size of the input layer” [Sheela and Deepa, 2013].

However, many opinions conclude to the fact that there is no perfect rule to define the optimal number of hidden nodes and proceed to the adoption of trial-and-error methods starting from the lowest possible number of nodes and gradually increasing it until the lowest error rate is achieved [Yuan et al., 2003]. After that point, the error increases anew. Moreover, difference in the performance between the training, validation and testing results also increases. While the perceptron achieves an excellent training performance rate, the error values in the testing or validation datasets are significantly higher (overfitting). Other authors adopt a pruning approach, following the inverse procedure, i.e. beginning with a relatively high number of neurons and gradually reducing it [Heaton, 2008], [Augasta and Kathirvalavakumar, 2013].

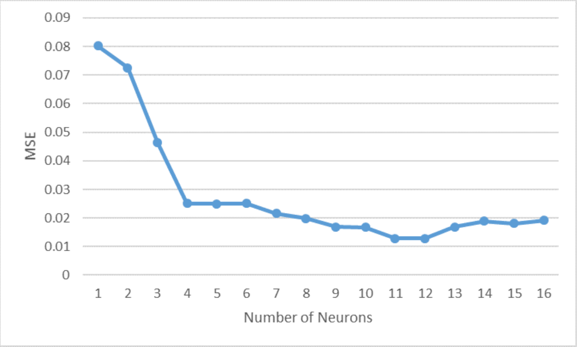

Figure 4.12: Average MSE per number of neurons - BPN SMS

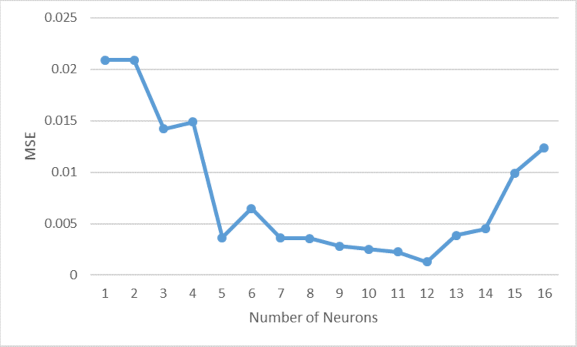

Figure 4.13: Average MSE per number of neurons - BPN calls

$$MSE = \frac{1}{m}\sum_{p=1}^{m} e_p^2 = \frac{1}{m}\sum_{p=1}^{m} (t_p - o_p)^2 \tag{4.42}$$

In the current chapter, we began experimenting in both the calls and SMS datasets with the lowest number of hidden neurons and measured the performance rate for each step. Mean Square Error (MSE), the mean value of the squared error e_p^2 for all the patterns of a dataset p, is the difference between the expected and the actual outputs the perceptron produces and constitutes its performance indicator. A formal depiction of the MSE is shown in Equation 4.42, where m is the total number of patterns, t the vector corresponding to the target values and o the vector indicating the actual values the perceptron produced.

Once the measurement phase was concluded, the decline in the MSE values was observed and the point where overfitting effects started appearing was encountered. Experiments were carried out for both plain backpropagation and pattern recognition perceptrons for every use case, data type and training algorithm. However, since the results were equal for all the different setups, the ones presented in the manuscript represent the whole picture. Figures 4.10 and 4.11 show the average plain backpropagation perceptron MSE values per number of neurons of the calls and SMS Datasets, whereas Figures 4.13 and 4.12

depict the average backpropagation pattern recognition perceptron MSE values per number of neurons of the Calls and SMS Datasets. Both the experimental setups ran the Levenberg-Marquardt backpropagation algorithm. Despite some increase exceptions, all the Figures indicate that the MSE rate decreases steeply after the second to third neuron addition and then decrease gradually until the twelfth neuron, where the lowest value is observed. From the thirteenth neuron and above, all the rates increase anew.

As a result of the aforementioned procedure, the plain and pattern recognition backpropagation perceptrons will carry twelve nodes in their hidden layer and the experimental phase will be analytically described in the next Chapter.

4.4.2 Evaluation

One of the issues that we faced during this phase was the decision upon a common evaluation method. The plain backpropagation perceptron and ANFIS are by default regression models, whereas the backpropagation pattern recognition perceptron solves classification problems. However, the fact that the regression model outputs receive approximate values to the ones present in the ground truth set $S_G = \{0.15, 0.25, 0.5, 0.75, 1\}$ allow for a more detailed classification, in order to figure out if the produced outcomes match the expected ones. In this line of reasoning, and in order to maintain uniform results, the classification algorithms from subsection 3.6 were used as means of comparison between the perceptrons' outputs and the ground truth values. The evaluation procedure is completed with the selection of the most efficient perceptron type.

4.4.3 Testing on Unknown Data

Once the perceptron with the best overall performance is identified, the following step is its test run on entirely unknown data. For that purpose, we performed a series of experiments on a Samsung Galaxy Ace 2 (GT-I8160) device, which was used for six consecutive months. The device was running the Android 4.4 version. Android Data Acquisition and Examination Tool (ADAET), a Python script, was implemented so as to extract the appropriate databases, perform the pre-processing, invoke the Matlab scripts for the NN testing and calculate the equivalent metrics.

ADAET initially establishes an ADB connection between the target device and a workstation. After verifying that the device is rooted, the *mmssms.db* and *calls.db* databases are copied and saved at the workstation. However, the data in their raw form are not in the appropriate format to be processed by the NN. Afterwards, they are pre-processed by the algorithm presented in Subsection 4.3.3. At a next step, ADAET invokes the Matlab scripts written for the plain backpropagation perceptron in Section 4.4 and lastly, the regression accuracy

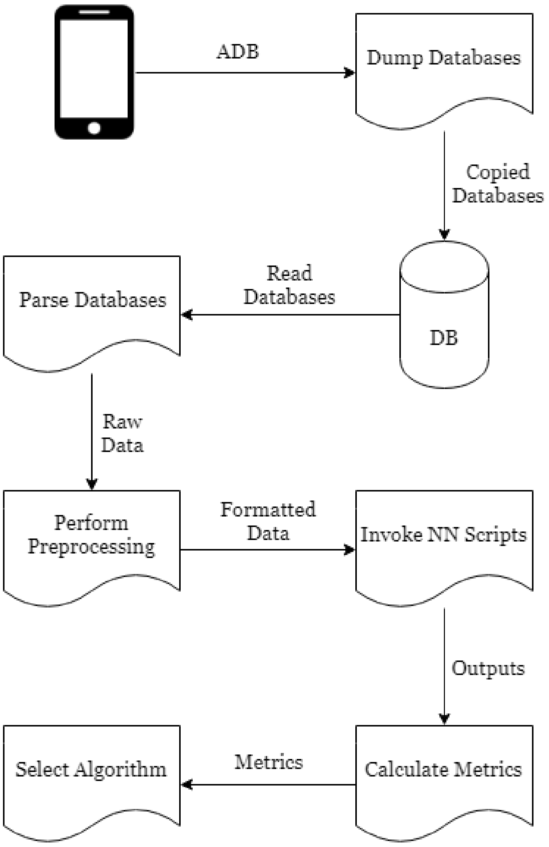

Figure 4.14: ADAET Functionality, extended from Spreitzenbarth and Uhrmann [Spreitzenbarth and Uhrmann, 2015]

and other metrics are calculated. The aforementioned procedure is depicted in Figure 4.14.

Once the definition of the methodology is complete, the aforementioned steps are followed towards the results generation, that are presented analytically in Chapter 5.

4.5 Summary of the Chapter

The current chapter provided a theoretical background to NNs and ANFIS, where their basic principles were elaborated. Afterwards, the proposed methodology of Chapter 2 was adapted to the needs of each technique. Taking into consideration that NNs and ANFIS, contrary to Fuzzy Systems, have memory and learning capabilities, the phase of Testing on unknown data was thereby applied and its mechanism was further elaborated. The following chapter will present the results of the experimental procedures for all the methods introduced in the current thesis.

The outcomes of this chapter include the following publication:

- Barmpatsalou, K., Cruz, T., Monteiro, E. and Simoes, P. (2018). Mobile Forensic Data Analysis: Suspicious Pattern Detection in Mobile Evidence, *IEEE Access.* Impact factor: 3.55.

Chapter 5

Results

...but beautiful mosaics are made of broken pieces.

(Lori J. Nelson)

This chapter presents the results that were generated after the conclusion of the experimental processes of Chapters 3 and 4. Initially, the results from the Fuzzy Systems experiments are presented. Afterwards, the results per use case and perceptron type for the CDA (training) dataset are provided. They are evaluated and the best performing perceptron and algorithm are selected. Once the aforementioned procedure is complete, ADAET acquires and processes calls and SMS data from the experimental mobile device and tests the performance metrics of the previously trained perceptron.

5.1 Fuzzy Systems Results

The Fuzzy Systems evaluation procedure includes the presentation of the results for each of the three datasets presented in Chapter 3.

Tables 5.1, 5.2 and 5.3 contain the cumulative results for all the candidate membership functions and their respective metrics after every classification type. After evaluating all the three datasets, the following observations were made:

- Triangular and Trapezoidal membership functions perform worse than the rest of the other candidates in every dataset and under every classification algorithm.

- The Bell membership function shows the best performance rates in every dataset; in the third dataset, its performance is equal to the one of the Gauss2 membership function.

M.F.	Algorithm	AUC	Accuracy	Precision	Recall	FPR
Triangular	kNN	0.583	0.267	0.811	0.267	0.175
	SVM	0.578	0.809	0.800	0.809	0.169
	Naive Bayes	0.567	0.805	0.649	0.805	0.174
	AdaBoost	**0.592**	**0.815**	**0.842**	**0.815**	**0.164**
	Random Forest	0.592	0.814	0.840	0.814	0.164
Trapezoidal	kNN	0.573	0.808	0.799	0.808	0.172
	SVM	0.573	0.808	0.799	0.806	0.172
	Naive Bayes	0.561	0.802	0.648	0.802	0.176
	AdaBoost	**0.574**	**0.808**	**0.846**	**0.808**	**0.171**
	Random Forest	**0.574**	**0.808**	**0.846**	**0.808**	**0.171**
Bell	kNN	0.923	0.951	0.951	0.9512	0.029
	SVM	0.748	0.824	0.825	0.824	0.102
	Naive Bayes	0.904	0.872	0.910	0.872	0.035
	AdaBoost	**0.974**	**0.981**	**0.981**	**0.981**	**0.009**
	Random Forest	0.945	0.963	0.964	0.963	0.021
Gauss	kNN	0.908	0.952	0.952	0.952	0.037
	SVM	0.858	0.864	0.889	0.864	0.058
	Naive Bayes	0.858	0.852	0.880	0.852	0.055
	AdaBoost	**0.925**	**0.960**	**0.961**	**0.960**	**0.030**
	Random Forest	0.915	0.956	0.956	0.956	0.032
Gauss2	kNN	0.924	0.961	0.961	0.961	0.0299
	SVM	0.884	0.871	0.903	0.871	0.0481
	Naive Bayes	0.882	0.865	0.893	0.865	0.0450
	AdaBoost	0.926	0.963	0.963	0.963	0.0305
	Random Forest	**0.931**	**0.963**	**0.963**	**0.963**	**0.0276**

Table 5.1: Evaluation metrics per membership function for the SMS Dev. 1 dataset

M.F.	Algorithm	AUC	Accuracy	Precision	Recall	FPR
Triangular	kNN	0.888	0.864	0.885	0.864	0.045
	SVM	0.875	0.822	0.840	0.822	0.052
	Naive Bayes	0.791	0.740	0.691	0.740	0.078
	AdaBoost	**0.897**	**0.850**	**0.870**	**0.850**	**0.043**
	Random Forest	0.890	0.867	0.888	0.867	0.045
Trapezoidal	**kNN**	**0.801**	**0.665**	**0.850**	**0.665**	**0.082**
	SVM	0.587	0.514	0.307	0.514	0.168
	Naive Bayes	0.727	0.684	0.606	0.684	0.107
	AdaBoost	0.742	0.704	0.647	0.704	0.102
	Random Forest	0.741	0.703	0.646	0.703	0.102
Bell	kNN	0.984	0.980	0.977	0.980	0.005
	SVM	0.976	0.968	0.966	0.968	0.008
	Naive Bayes	0.846	0.809	0.743	0.809	0.054
	AdaBoost	**0.998**	**0.997**	**0.997**	**0.997**	**0.001**
	Random Forest	0.991	0.989	0.986	0.989	0.004
Gauss	kNN	0.987	0.984	0.982	0.984	0.004
	SVM	0.980	0.972	0.9709	0.972	0.007
	Naive Bayes	0.850	0.815	0.746	0.815	0.052
	AdaBoost	**0.995**	**0.994**	**0.991**	**0.994**	**0.001**
	Random Forest	0.991	0.989	0.986	0.989	0.002
Gauss2	kNN	0.986	0.983	0.981	0.983	0.004
	SVM	0.988	0.984	0.982	0.984	0.003
	Naive Bayes	0.880	0.848	0.781	0.848	0.040
	AdaBoost	**0.989**	**0.986**	**0.983**	**0.986**	**0.003**
	Random Forest	0.988	0.984	0.982	0.984	0.003

Table 5.2: Evaluation metrics per membership function for the SMS Dev. 2 dataset

M.F.	Algorithm	AUC	Accuracy	Precision	Recall	FPR
Triangular	kNN	0.619	0.310	0.857	0.310	0.158
	SVM	0.611	0.582	0.508	0.582	0.159
	Naive Bayes	0.604	0.573	0.365	0.573	0.160
	AdaBoost	**0.617**	**0.591**	**0.651**	**0.591**	**0.156**
	Random Forest	0.617	0.590	0.610	0.590	0.157
Trapezoidal	kNN	0.608	0.294	0.571	0.294	0.143
	SVM	0.609	0.294	0.571	0.294	0.143
	Naive Bayes	0.600	0.571	0.365	0.571	0.162
	AdaBoost	**0.606**	**0.579**	**0.371**	**0.579**	**0.160**
	Random Forest	0.605	0.578	0.371	0.579	0.161
Bell	kNN	0.971	0.963	0.963	0.962	0.010
	SVM	0.937	0.906	0.922	0.906	0.025
	Naive Bayes	0.722	0.682	0.527	0.682	0.102
	AdaBoost	**0.990**	**0.986**	**0.986**	**0.986**	**0.004**
	Random Forest	0.983	0.978	0.978	0.978	0.033
Gauss	kNN	0.979	0.971	0.972	0.971	0.008
	SVM	0.940	0.909	0.975	0.975	0.025
	Naive Bayes	0.713	0.666	0.519	0.666	0.191
	AdaBoost	**0.990**	**0.986**	**0.986**	**0.986**	**0.006**
	Random Forest	0.981	0.975	0.975	0.975	0.006
Gauss2	**kNN**	**0.975**	**0.967**	**0.968**	**0.967**	**0.009**
	SVM	0.944	0.915	0.931	0.915	0.023
	Naive Bayes	0.716	0.671	0.521	0.671	0.108
	AdaBoost	0.949	0.920	0.935	0.920	0.022
	Random Forest	0.946	0.917	0.932	0.917	0.022

Table 5.3: Evaluation metrics per membership function for the SMS Dev. 3 dataset

- In the majority of the tests, the AdaBoost and Random Forest classification algorithms showed the best performance rates. On the contrary, kNN, SVM and Naive Bayes showed the poorest performance.

- The performance difference among the Bell, Gauss and Gauss2 membership function is very low and they can be considered as efficient alternatives.

The evaluation procedure proved that Fuzzy Systems can successfully detect and categorize patterns according to their degree of suspiciousness in small-scale problems. It constituted a proof-of-concept for the efficiency of intelligent computation techniques. However, when the input space and subsequently the complexity of the system itself increase, the particular solution faces scalability issues. The next section attempts to solve the particular problem, by presenting the results of the methodology application to Neural Networks and the Adaptive Neuro-Fuzzy Inference System (ANFIS).

5.2 Neural Networks and ANFIS Results

The first part of the results presentation is associated to the performance evaluation of three different perceptron types (plain, backpropagation and

ANFIS) for calls and SMS data, per corresponding use case. Firstly, the results for the CDA datasets are presented

5.2.1 CDA Dataset Results

This subsection is divided in two separate parts, that present and discuss the results of the Cyberbullying and Drug Dealing use cases.

5.2.1.1 Cyberbullying

This section is divided in three parts; the first comprises the results for the plain backpropagation perceptron, the second for the pattern recognition backpropagation perceptron and the third for ANFIS.

Plain Backpropagation Perceptron

For the calls dataset, the training algorithms showed satisfactory performance, with accuracy scores varying from 87.5 to 97.4%. The Levenberg-Marquardt and Bayesian Regularization algorithms outperform the rest, with the latter achieving the highest Accuracy score. In terms of Precision and Recall, only Bayesian Regularization scores over 80% and Levenberg-Marquardt follows with results in the mid-70s range. The rest of the algorithms have rather poor or unbalanced rates. More details can be found in Table 5.4, whereas Fig. 5.1 depicts the aforementioned metrics per algorithm.

For the SMS dataset, all the training algorithms scored within the 71.3-92.4% regression accuracy spectrum, that constitutes a fair to very good performance, but lower than the Calls dataset equivalents. Levenberg-Marquardt, Bayesian Regularization and BFGS Quasi-Newton backpropagation achieved the higher Accuracy scores (>80%), whereas the Gradient Descent family of algorithms showed the poorest performance. Levenberg-Marquardt backpropagation shows significantly higher Precision scores for each suspiciousness category. The Recall rates are slightly lower and all the algorithms perform equally. Table 5.5 shows the performance metrics for the SMS dataset,

Backpropagation Pattern Recognition Neural Network

Performance for the Pattern Recognition perceptron is not as uniform as the Plain Backpropagation perceptron‘s. Accuracy for the Calls training, validation and testing datasets varies from a minimum 41.6% to a maximum 99.8%. However, only the validation and testing subsets are taken higher into consideration. Similarly to the results of the previous subsection, the Levenberg-Marquardt and Bayesian Regularization outperform the rest of the algorithms. Nevertheless, Levenberg-Marquardt shows a more balanced profile

Table 5.4: Plain Backpropagation Perceptron Performance - Cyberbullying: Calls

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.00179	92.4	77.5	74.4
Bayesian Regularization	0.000341	97.4	80.9	82.6
BFGS Quasi-Newton	0.00461	91.1	68.9	57.2
Scaled Conjugate Gradient	0.00398	91.1	75.4	72.7
One-Step Secant	0.00874	89.2	67.4	57.1
Resilient	0.00504	89.9	59.4	55.4
G. D.	0.0167	90.0	71.4	56.7
G. D. Momentum	0.0306	87.5	49.1	52.3
G. D. Momentum-Adaptive	0.0081	91.3	76.3	66.3
C. G. Powell-Beale	0.00427	90.8	66.3	59.6
C. G. Polak-Ribière	0.00392	91.4	76.6	49.7
C. G. Fletcher-Reeves	0.00428	89.9	69.7	63.7

Table 5.5: Plain Backpropagation Perceptron Performance - Cyberbullying: SMS

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.00277	92.4	87.7	71.4
Bayesian Regularization	0.00034	83.7	74.3	72.1
BFGS Quasi-Newton	0.00461	81.6	75.9	73.5
Scaled Conjugate Gradient	0.00398	78.5	74.3	71.4
One-Step Secant	0.00874	78.4	72.7	71.3
Resilient	0.00504	79.7	73.5	71.0
G. D.	0.0167	71.3	70.4	67.7
G. D. Momentum	0.0306	72.5	71.3	70.5
G. D. Momentum-Adaptive	0.0081	71.4	62.7	60.9
C. G. Powell-Beale	0.00427	73.5	64.0	61.7
C. G. Polak-Ribière	0.00392	78.3	70.0	67.6
C. G. Fletcher-Reeves	0.00428	79.6	77.0	73.3

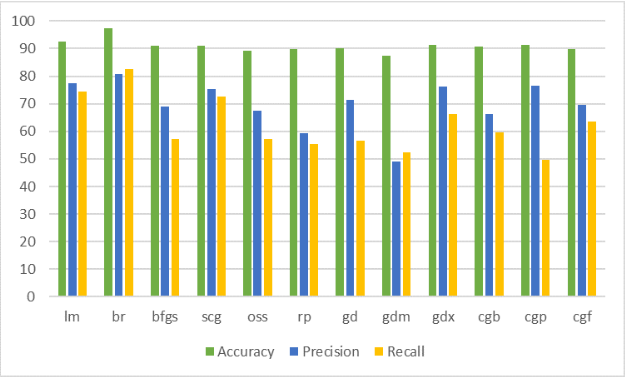

Figure 5.1: Performance histogram for the plain backpropagation perceptron - Cyberbullying: Calls

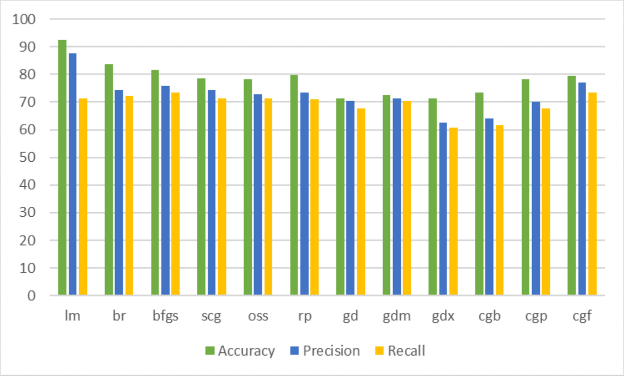

Figure 5.2: Performance histogram for the plain backpropagation perceptron - Cyberbullying: SMS

Table 5.6: Backpropagation Pattern Recognition Perceptron Performance - Cyberbullying: Calls

Algorithm	**Perf.**	**Training**	**Validation**	**Testing**
Levenberg-Marquardt	0.00998	95.4	93.2	94.9
Bayesian Regularization	0.000442	99.8	-	94.0
BFGS Quasi-Newton	0.0761	86.1	87.6	87.2
Scaled Conjugate Gradient	0.0690	89.4	87.6	87.2
One-Step Secant	0.0933	83.0	81.3	84.4
Resilient	0.0768	87.3	84.4	88.3
Gradient Descent (G.D.)	0.234	45.0	41.6	46.0
G. D. Momentum	0.229	54.9	57.9	56.6
G. D. Momentum-Adaptive	0.0953	83.0	84.4	83.5
C. G. Powell-Beale	0.05846	90.8	87.4	88.3
C. G. Polak-Ribière	0.0738	86.9	84.4	87.4
C. G. Fletcher-Reeves	0.0867	83.0	85.5	82.6

between training, testing and validation, whereas Bayesian Regularization, that by default lacks a validation dataset, presents a declining of almost six points. Once again, the Gradient Descent algorithms other than the variation with momentum and adaptive learning show the worst performance results. A more detailed overview is provided in Table 5.6.

The declining between the performance rates is smaller for the SMS dataset. The accuracy range is defined between 81.2% and 96.3%, with the Levenberg-Marquardt and Bayesian Regularization algorithms scoring the highest numbers anew. The difference between the rest of the algorithms is insignificant and only the simple Gradient Descent variation shows the lowest scores. Table 5.7 presents the respective results.

ANFIS

Due to the big number of inputs and corresponding linguistic variable subdivisions, it was impossible to create fuzzy systems manually (Type-1) by generating them from the data (Type-2). As a result, fuzzy clustering was the only available option in order to create the input space. Modifications in

Table 5.7: Backpropagation Pattern Recognition Perceptron Performance - Cyberbullying: SMS

Algorithm	Perf.	Training	Validation	Testing
Levenberg-Marquardt	0.0126	96.3	94.8	95.4
Bayesian Regularization	0.0166	95.6	-	92.6
BFGS Quasi-Newton	0.0478	90.1	91.5	90.6
Scaled Conjugate Gradient	0.0445	91.6	88.9	89.7
One-Step Secant	0.0454	91.7	92.3	88.0
Resilient	0.0347	92.7	94.0	88.9
G. D.	0.111	86.1	87.2	81.2
G. D. Momentum	0.145	84.8	88.0	89.7
G. D. Momentum-Adaptive	0.0412	91.9	91.5	92.3
C. G. Powell-Beale	0.0436	89.9	94.0	90.6
C. G. Polak-Ribière	0.0529	89.2	86.3	85.5
C. G. Fletcher-Reeves	0.0346	91.7	90.6	87.2

Table 5.8: ANFIS Performance - Cyberbullying: Calls

Version	1	2	3	4	5
Range of Influence	0.5	0.25	0.75	0.45	0.45
Squash Factor	1.25	1.25	1.25	1.15	1.05
M.F.s	20	30	18	24	27
Error	0.0661	0.0768	0.067	0.0604	0.0549
Accuracy	90.9	91.3	91.4	89.9	90.1
Precision	43.8	51.5	42.3	48.1	30.2
Recall	30.4	37.0	23.9	28.3	28.3

the squash factor and range of influence values resulted in different numbers of membership functions.

Despite the variations among the number of membership functions per instance, the difference between the Error and Accuracy rates do not surpass 2% for the Calls dataset. Moreover, despite the rather satisfactory average Accuracy percentages scored, the Precision and Recall rates are rather low. The amount of membership functions was between 18 and 30, whereas the version with the best overall performance in terms of Accuracy, Precision and Recall is the second column of Table 5.8.

Similar conclusions can be extracted from the SMS dataset, where the difference between the highest and the lowest Error and Accuracy values is not greater than 3.5%. The higger amount of membership functions generated was 57, whereas the lower was 18. The Accuracy scores are slightly higher than the ones achieved for the Calls dataset. However, the Precision and Recall metrics are significantly higher, but yet not within the acceptable rates for a very good performance. More details about the ANFIS performance of the SMS dataset can be encountered in Table 5.9. The version with the best performance rates can be found in the fourth column of the aforementioned table.

The next subsection delves into the results generated by the three different perceptrons for the Drug Dealing use case and provides more information that will lead to the appropriate method selection.

Table 5.9: ANFIS Performance - Cyberbullying: SMS

Version	**1**	**2**	**3**	**4**	**5**
Range of Influence	0.5	0.25	0.75	0.4	0.4
Squash Factor	1.25	1.25	1.25	1.15	1.05
M.F.s	22	57	13	30	38
Error	0.075	0.097	0.085	0.072	0.068
Accuracy	90.6	91.8	88.6	91.9	92.0
Precision	58.1	61.5	56.1	62.4	62.5
Recall	59.1	62.7	57.4	63.3	63.3

5.2.1.2 Drug Dealing

The structure of the current section follows the pattern of section 5.2.1.1, where each part shows the performance evaluation results for each perceptron type.

Plain Backpropagation Perceptron

The results concerning the Calls dataset of the Drug Dealing use case show rather high Accuracy rates within the 83.7-98.2% spectrum and are also accompanied by excellent Precision and Recall metrics of the 90s scale, at least for the best performing algorithms. Similarly to the Cyberbullying use case, Levenberg-Marquardt and Bayesian Regularization showed the best performance rates in all the metric categories. Conjugate Gradient with Fletcher-Reeves Updates and Resilient backpropagation followed with almost equivalently high Accuracy rates, but relatively lower Precision and Recall scores. The lowest performance score was achieved by the Gradient Descent and Gradient Descent with Momentum algorithms. More details about the performance of the algorithms are depicted in Table 5.10 and in Fig. 5.3.

Similar results were encountered in the SMS dataset, the Accuracy of which, however, covered a broader area of ranges, varying from 66.6% to 97.8%. Moreover, the Precision and Recall metrics were the highest out of all the datasets for the plain backpropagation perceptron experiments. This is the only dataset where the Levenberg-Marquardt algorithm did not have one of the two first places in performance, but achieved the third best scores after Bayesian Regularization and Conjugate Gradient with Polak-Ribière Updates. Gradient Descent and Gradient Descent with Momentum showed once again poor results. Table 5.11 and Fig. 5.4 analytically present the perceptron results for the SMS dataset.

Backpropagation Pattern Recognition Perceptron

The current use case and dataset is an example of a non-successfully concluded experimental setup. All the algorithms failed to classify almost or more than half of the patterns of different suspiciousness for the Calls dataset. Both the

Table 5.10: Plain Backpropagation Perceptron Performance - Drug Dealing: Calls

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.0275	96.2	89.2	94.3
Bayesian Regularization	0.00230	98.2	96.6	96.6
BFGS Quasi-Newton	0.0111	94.6	84.5	68.2
Scaled Conjugate Gradient	0.00613	92.9	70.9	63.6
One-Step Secant	0.00798	92.0	73.1	77.3
Resilient	0.00875	95.7	78.1	93.2
G. D.	0.0373	89.0	57.3	53.4
G. D. Momentum	0.0440	83.7	49.4	45.5
G. D. Momentum-Adaptive	0.00974	95.6	79.8	89.8
C. G. Powell-Beale	0.00605	92.4	73.8	54.5
C. G. Polak-Ribière	0.0102	93.4	74.4	69.3
C. G. Fletcher-Reeves	0.00538	96.7	88.0	92.0

Table 5.11: Plain Backpropagation Perceptron Performance - Drug Dealing: SMS

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.00123	96.5	95.3	96.2
Bayesian Regularization	0.000827	97.8	98.4	96.7
BFGS Quasi-Newton	0.0025	96.1	93.6	96.8
Scaled Conjugate Gradient	0.00217	95.6	92.6	96.6
One-Step Secant	0.00265	92.3	87.6	92.8
Resilient	0.00213	94.9	89.9	97.8
G. D.	0.0229	66.6	56.6	56.2
G. D. Momentum	0.0338	74.7	66.1	76.0
G. D. Momentum-Adaptive	0.00467	89.9	84.4	89.4
C. G. Powell-Beale	0.00255	93.3	89.4	92.9
C. G. Polak-Ribière	0.00221	96.6	96.0	95.9
C. G. Fletcher-Reeves	0.00191	97.0	96.8	96.0

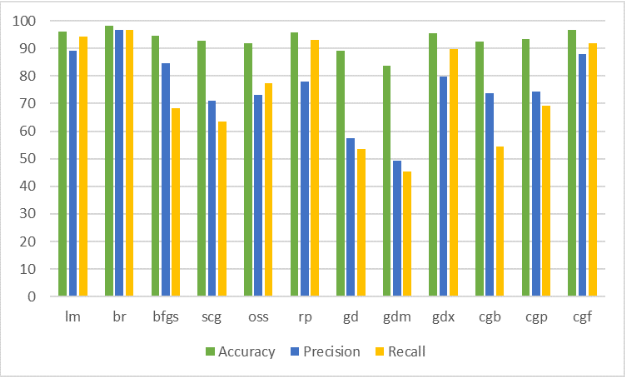

Figure 5.3: Performance histogram for the plain backpropagation perceptron - Drug Dealing: Calls

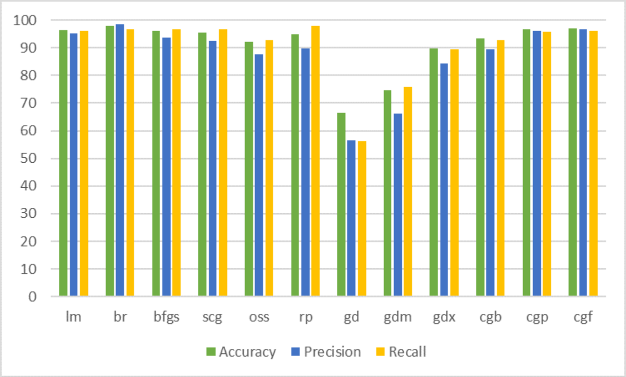

Figure 5.4: Performance histogram for the plain backpropagation perceptron - Drug Dealing: SMS

Table 5.12: Backpropagation Pattern Recognition Perceptron Performance - Drug Dealing: Calls

Algorithm	**Perf.**	**Training**	**Validation**	**Testing**
Levenberg-Marquardt	0.111	56.2	47.4	51.1
Bayesian Regularization	0.116	54.7	-	57.1
BFGS Quasi-Newton	3.39	52.2	55.6	62.4
Scaled Conjugate Gradient	3.33	52.5	55.6	59.4
One-Step Secant	3.16	54.3	51.9	50.4
Resilient	3.17	54.6	52.6	53.4
G. D.	3.44	43.1	40.6	44.4
G. D. Momentum	3.34	41.7	41.4	47.4
G. D. Momentum-Adaptive	3.36	52.2	53.4	63.2
C. G. Powell-Beale	3.25	53.6	54.9	55.6
C. G. Polak-Ribière	3.27	53.5	57.9	53.4
C. G. Fletcher-Reeves	3.19	54.8	52.6	51.9

training, validation and testing sessions did not provide an Accuracy score over 65%. The BFGS Quasi-Newton and Gradient Descent with Momentum and Adaptive Learning Rate performed slightly better than the rest of the algorithms, but the remaining members of the Gradient Descent family showed the worst results. More details about the scoring can be found in Table 5.12.

Contrary to the Calls dataset, the SMS dataset showed excellent Accuracy results that reached up to 99.7% for the training subset and 99.1% for the validation and testing equivalents. The Levenberg-Marquardt and Bayesian Regularization algorithms outperformed the rest and the lowest scores were marked for the Gradient Descent and Gradient Deswcent with Momentum algorithms. Table 5.13 presents the respective results.

ANFIS

Five different ANFIS versions were produced for each dataset. As far as the Calls dataset is concerned, the total amount of membership functions generated varied from 18 to 30. Despite the variety between their numbers, the Accuracy

Table 5.13: Backpropagation Pattern Recognition Perceptron Performance - Drug Dealing: SMS

Algorithm	**Perf.**	**Training**	**Validation**	**Testing**
Levenberg-Marquardt	0.000820	99.7	99.1	99.1
Bayesian Regularization	0.000899	99.7	-	98.3
BFGS Quasi-Newton	0.024	96.7	95.5	95.5
Scaled Conjugate Gradient	0.0146	97.3	97.0	96.8
One-Step Secant	0.0241	96.7	97.6	95.9
Resilient	0.0152	97.1	96.3	96.7
G. D.	0.158	72.8	75.7	73.7
G. D. Momentum	0.111	81.7	80.5	80.7
G. D. Momentum-Adaptive	0.0156	97.4	97.0	96.8
C. G. Powell-Beale	0.0161	96.9	97.8	95.7
C. G. Polak-Ribière	0.0146	97.3	97.4	96.8
C. G. Fletcher-Reeves	0.0174	97.2	96.1	97.6

Table 5.14: ANFIS Performance - Drug Dealing: Calls

Version	**1**	**2**	**3**	**4**	**5**
Range of Influence	0.5	0.25	0.75	0.45	0.25
Squash Factor	1.25	1.25	1.25	1.25	0.8
M.F.s	19	30	18	24	27
Error	0.098	0.0458	0.0525	0.0453	0.042
Accuracy	95.3	95.7	95.5	96.5	96.5
Precision	81.4	83.9	81.0	86.3	86.4
Recall	89.8	88.6	92.0	93.2	93.5

metrics were all similar and scored in the interval of 95.3 - 96.5%. Contrary to the cyberbullying use case, the Precision and Recall metrics were relatively high, over 80%. The best performance level was achieved by the fifth version, consisting of 27 membership functions. Analytical details are provided in Table 5.14.

Similar, but borderline lower performance was noted for the SMS dataset. The number of membership functions per version varied from 15 to 51 and the Accuracy scores from 86.3% to 93.2%. As it can be observed in Table 5.15, despite the considerable increase in the number of membership functions between the best performing versions 2 and 5, the subsequent increase the performance score is very low. A fuzzy system with 51 membership functions is computationally slower than a version equipped with 26 and since the performance difference is rather low, the version with the lower number of membership functions can be selected as the most efficient in terms of performance and computational cost.

The aforementioned subsections showed that a decision upon the best approach for the identification of suspicious and non-suspicious patterns in mobile metadata is not a simplified procedure. However, there are some characteristics that clarify the selection procedure and are analytically presented in the following paragraphs.

Table 5.15: ANFIS Performance - Drug Dealing: SMS

Version	**1**	**2**	**3**	**4**	**5**
Range of Influence	0.5	0.25	0.75	0.45	0.45
Squash Factor	1.25	1.25	1.25	1.15	1.05
M.F.s	15	51	10	22	26
Error	0.0665	0.0521	0.071	0.0622	0.0578
Accuracy	89.8	93.2	86.3	88.1	92.0
Precision	88.0	90.0	87.1	84.3	88.0
Recall	86.0	92.9	76.6	85.3	91.8

5.2.2 Best Performance Perceptron and Algorithm Selection

Defining the most appropriate NN perceptron for the detection and rating of suspicious patterns is a rather complicated process, especially when the majority of the produced results are equivalently good. In such a case, the selection criteria are not limited to the success rates of each method, but focus on deeper levels of detail.

Generally, all the three perceptron types achieved a relatively high average performance rate, especially for their top variations. Accuracy rates over 80% were a common characteristic. Despite its satisfactory performance in three out of the four dataset and use case combination, the pattern recognition backpropagation perceptron performed significantly lower than expected for the Calls dataset of the Drug Dealing use case. On the contrary, the plain backpropagation perceptron and ANFIS did not face a similar issue. ANFIS showed an excellent performance profile for the Drug Dealing use case, but the Precision and Recall rates for the Cyberbullying use case were rather low. The plain backpropagation perceptron was the most stable method out of the three. Its results might not have reached the rates generated by the pattern recognition perceptron, but it was able to maintain a uniform average of results, especially for the best performing family of algorithms. As a result, the plain backpropagation perceptron is considered the preferred approach for solving the suspicious pattern detection problem from mobile forensic data.

Additionally, the observation and selection procedure also resulted in three noteworthy conclusions. Firstly, ANFIS is highly dependent on the amount of patterns under examination. Tables 5.8, 5.9, 5.14 and 5.15 indicate that the upper bound of the produced membership functions is significantly higher for the SMS than the Calls datasets. This can be justified by the fact that the amount of patterns in the SMS dataset is almost six times bigger than the Calls equivalent, as seen in Table 4.5. This observation though brings a scalability issue to the surface. As already mentioned in the previous section, ANFIS versions with many membership functions come at a high computational cost. Consequently, the ANFIS problem solving capability is finite and its performance versus efficiency ratio drops as the number of the patterns in the input space increases.

Secondly, the regression approach of the plain backpropagation perceptron is more efficient than the classification approach of the pattern recognition backpropagation network. The difference between the plain backpropagation and pattern recognition backpropagation perceptrons performance for the Calls dataset of the Drug Dealing use case is substantially considerable. While the former was able to detect most of the patterns correctly, the latter failed at the classification of a little less than 50%. This statement is not useful as a standalone assumption. However, if Table 4.5 is taken into consideration, it is noticeable that the specific dataset has a less proportional pattern distribution than the remaining ones. Moreover, it has a smaller amount of total patterns when compared to the SMS datasets. The aforementioned results signify that the plain backpropagation perceptron is more capable of correctly detecting patterns with uneven distribution, fact that renders it more suitable as a tool for real-life circumstances.

Lastly, as far as the backpropagation algorithms are concerned, the Levenberg-Marquardt and Bayesian Regularization methods showed by far the best results. On the one hand, Bayesian Regularization achieved higher performance rates but showed a considerable amount of difference between training and testing datasets. On the other hand, the Levenberg-Marquardt algorithm showed moderately lower performance rates, but maintained the result uniformity. The difference between the aforementioned algorithms and the remaining ones was remarkably observable. Conjugate Gradient with Fletcher-Reeves Updates and Resilient backpropagation provided satisfactory results, while the Scaled Conjugate Gradient and BFGS Quasi-Newton backpropagation algorithms follow with vaguely noticeable performance declining. Two members of the Gradient Descent family, simple Gradient Descent and its momentum variation had the worst performance rates for all the experimental setups.

Once the appropriate approach is selected, the research procedure continues with testing the plain backpropagation perceptron on completely unknown data that are previously acquired from a mobile device. This scenario is closer to real circumstances and will test if the perceptron and its respective algorithms‘ efficiency is aligned with the actual test results.

5.2.3 Testing on Unknown Data

The plain backpropagation perceptron showed overall better performance rates compared to the rest of the employed techniques. This section presents the behaviour of the previously trained perceptron with the CDA dataset patterns when entirely unknown data are used as inputs to the system. However, a limitation considering the pattern distribution needs to be taken into consideration beforehand.

Table 4.5 presents the occurrences of patterns, classified by their suspiciousness level, according to the ground truth generation. Since the device operated in real-life circumstances, the uniformity between the occurrences is significantly lower than the CDA dataset‘s. The pattern distribution has an effect on

Table 5.16: Samsung Device GT Pattern Distribution

Use Case Cat.	**Cyberbullying Calls**	**Cyberbullying SMS**	**Drug Dealing Calls**	**Drug Dealing SMS**
0.15	400	383	238	597
0.25	66	152	246	482
0.5	31	544	8	–
0.75	3	–	7	–
1	–	–	–	–
Total	500	1079	500	1079

Table 5.17: Samsung Device Backpropagation Perceptron Performance - Cyberbullying: Calls

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.1023	89.0	79.7	79.2
Bayesian Regularization	5.4039	88.8	79.6	80.0
BFGS Quasi-Newton	0.1026	89.0	76.8	84.3
Scaled Conjugate Gradient	0.6159	89.4	74.6	77.9
One-Step Secant	0.0393	89.0	84.2	68.0
Resilient	0.1125	90.0	71.7	78.7
G. D.	0.0850	88.0	81.3	85.7
G. D. Momentum	0.0473	88.4	68.5	76.1
G. D. Momentum-Adaptive	0.3622	89.0	77.5	76.0
C. G. Powell-Beale	0.4134	90.0	75.1	77.2
C. G. Polak-Ribière	0.1254	90.8	77.6	74.5
C. G. Fletcher-Reeves	0.0328	88.0	75.1	73.3

the calculation of the regression accuracy and the other metrics. As already mentioned in Section 4.4.2, the metrics are calculated by using 10-fold cross validation. However, when the number of patterns per category is less than 10, the respective metrics are omitted because the number of actual patterns is lower than the folds and no effective comparison can take place. It is also expected that the actual device datasets do not contain patterns of the highest suspiciousness level.

Table 5.17 presents the results for the Calls dataset of the Cyberbullying use case. The table constitutes an interesting case, because all the algorithms perform at approximately the same level, despite the differences encountered during the experimental phase in Section 5.2.1. Since the Accuracy metrics do not show significant differences, the Precision and Recall results will be examined. The Levenberg-Marquardt, Bayesian Regularization and BFGS Quasi-Newton algorithms have the most balanced performance. Surprisingly enough, in the specific sample, Gradient Descent shows a very efficient profile as well.

The performance of the SMS dataset is depicted in Table 5.18. The Levenberg-Marquardt algorithms performs significantly better than the rest of the backpropagation methods, with Bayesian Regularization and BFGS Quasi-Newton following closely. Bayesian Regularization shows a significant difference in its performance, when compared to the training and testing experimental phase. The Gradient Descent family of algorithms shows the poorest performance once again.

Table 5.18: Samsung Device Backpropagation Perceptron Performance - Cyberbullying: SMS

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.0203	92.4	94.6	85.4
Bayesian Regularization	0.0048	83.7	73.3	72.1
BFGS Quasi-Newton	0.0139	81.6	75.9	73.5
Scaled Conjugate Gradient	0.0153	78.5	74.2	71.3
One-Step Secant	0.0158	78.4	72.6	71.2
Resilient	0.0169	78.7	73.5	71.0
G. D.	0.0332	71.3	70.4	67.7
G. D. Momentum	0.0509	72.5	71.2	70.4
G. D. Momentum-Adaptive	0.0211	71.4	62.7	60.9
C. G. Powell-Beale	0.0176	73.5	64.0	61.7
C. G. Polak-Ribière	0.0164	78.3	70.0	67.6
C. G. Fletcher-Reeves	0.0142	79.6	76.6	75.7

Table 5.19: Samsung Device Backpropagation Perceptron Performance - Drug Dealing: Calls

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	0.383	85.4	82.4	89.4
Bayesian Regularization	0.8458	83.0	81.7	87.4
BFGS Quasi-Newton	0.0408	83.8	82.9	86.6
Scaled Conjugate Gradient	0.0729	83.6	83.8	86.2
One-Step Secant	0.1988	88.2	88.9	87.8
Resilient	0.1528	79.2	78.5	82.9
G. D.	0.1718	79.2	89.6	80.9
G. D. Momentum	0.2832	87.6	84.3	93.9
G. D. Momentum-Adaptive	0.407	81.2	79.1	86.2
C. G. Powell-Beale	0.19	85.8	86.8	85.8
C. G. Polak-Ribière	0.0375	81.4	80.6	82.9
C. G. Fletcher-Reeves	0.0602	80.4	80.6	82.5

Table 5.20: Samsung Device Backpropagation Perceptron Performance - Drug Dealing: SMS

Algorithm	**Perf.**	**Acc.**	**Prec.**	**Rec.**
Levenberg-Marquardt	5.084e-04	99.6	99.7	99.6
Bayesian Regularization	0.0116	99.9	99.8	99.8
BFGS Quasi-Newton	7.177e-04	99.9	99.8	99.8
Scaled Conjugate Gradient	1.432e-04	99.9	99.8	99.8
One-Step Secant	1.687e-04	99.8	99.7	99.6
Resilient	8.402e-04	93.2	94.9	89.6
G. D.	0.015	91.9	86.6	96.9
G. D. Momentum	0.0178	82,0	92.2	51.7
G. D. Momentum-Adaptive	0.0211	94.5	85.4	85.6
C. G. Powell-Beale	1.755e-04	99.7	99.6	99.4
C. G. Polak-Ribière	3.072e-04	99.9	99.8	99.8
C. G. Fletcher-Reeves	7.390e-04	99.5	99.5	99.0

The results deriving from the examination of Table 5.19 for the Calls dataset of the Drug Dealing use case constitute a performance surprise. Firstly, none of the algorithms does not have an Accuracy score over 90%. Secondly, other than the rather expected Levenberg-Marquardt best performance, Gradient Descent with Momentum shows the best results in the category, with its Recall levels reaching almost 94%. The rest of the results are also uniform, with small variations.

Finally, Table 5.20 shows the performance rates of the SMS dataset. The produced results are rather impressive, with almost excellent metrics. This happens partially because of the existence of only two patterns in the dataset space, as it can be inferred from Table 5.16. Almost all the algorithms, other than the Gradient Descent and its variation with Momentum performed equally well.

Figure 5.5 is a performance diagram of all the backpropagation algorithms for each use case and dataset of the actual device testing case. The last column of each diagram subsection is the average performance of every algorithm. The superiority of the Levenberg-Marquardt, Bayesian Regularization and BFGS Quasi-Newton backpropagation can be inferred directly from the diagram. In general, the algorithms did not show a different behaviour between the two different datasets. Levenberg-Marquardt showed the most balanced behaviour, whereas the performance differences for Bayesian Regularization were a bit more considerable. As a result, the most appropriate combination for examining a mobile device for suspicious patterns and classifying the total patterns in different categories is the use of plain backpropagation perceptrons, trained by either Levenber-Marquardt or Bayesian Regularization algorithms.

5.3 Summary of the Chapter

This chapter provided the results for the total of experimental processes that took place during the current research. The first section was dedicated to the results of the Fuzzy Systems use case scenario, whereas the second concerned

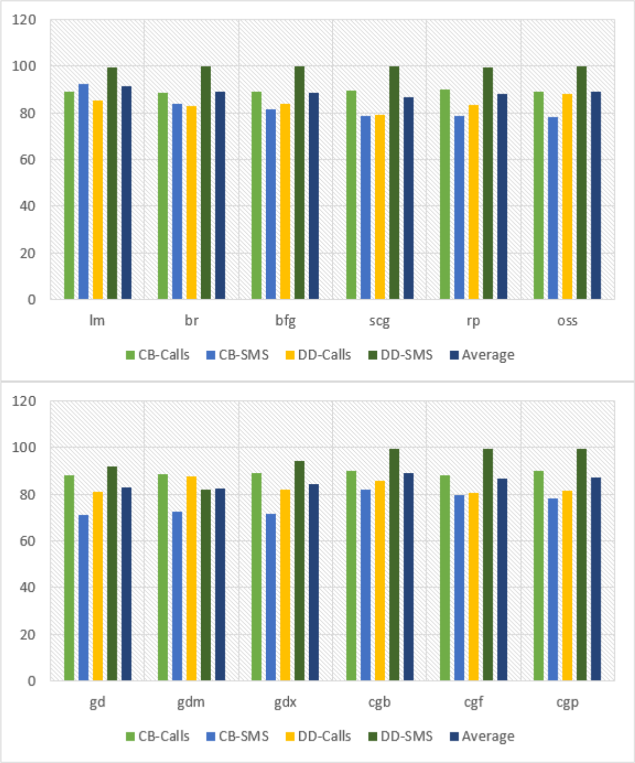

Figure 5.5: Average performance per training algorithm

the use of NNs and ANFIS for the cyberbullying and low-level drug dealing use case scenarios. The Fuzzy Systems results showed a higher level of overall stability and performance, but their scalability and learning incapability issues render them suitable only for limited-input scenarios. Moreover, the difference in the performance ratings between Fuzzy Systems, NNs and ANFIS is not an impediment that can lead to the rejection of the latter two techniques. NNs and ANFIS showed similar results, with both NN types (plain and backpropagation) requiring a less complicated configuration procedure than ANFIS. Lastly, the plain backpropagation perceptron showed a better result stability during all the experimental phases, when compared to the pattern recognition backpropagation perceptron. Despite the fact that our methodology was successfully tested on different scenarios, there are still some aspects that need to be addressed. The next and final Chapter of the thesis will provide more details on how the existing work can be expanded and further improved.

The outcomes of this chapter include were parts of the following submissions:

- Barmpatsalou, K., Cruz, T., Monteiro, E. and Simoes, P. (2018). Mobile Forensic Data Analysis: Suspicious Pattern Detection in Mobile Evidence, *IEEE Access*. Impact factor: 3.55.

Chapter 6

Conclusions and Future Work

> Cliffs are meant to be awe inspiring, and respected. Cliffs were also meant to be scaled and conquered.
>
> *(Anthony T. Hincks)*

This final chapter provides an overview of the thesis and its contributions. Moreover it discusses the findings of the current research and enlists future work potentials.

6.1 Synthesis of the Thesis

Chapter 2 introduced some of the more important principles of Mobile Forensics (MF), presented the research trends and identified that Mobile Forensic Data Analysis (MFDA) is one of the concepts in need of further evolution. Furthermore, it performed a bibliographic analysis on digital criminal profiling methodologies and introduced the one that consisted the backbone of the current thesis.

Fuzzy Systems, due to their simple inference mechanism and the relatively comprehensible functionality, served as a proof of concept for a simple suspicious pattern identification scenario from SMS data and metadata. In Chapter 3, the case of Public Protection and Disaster Relief (PPDR) officers infiltrating the rioting forces was examined and the respective methodology adaptations were presented.

Chapter 4 examined two more use case scenarios, by making use of Neural Networks (NNs) and the Adaptive Neuro-Fuzzy Inference System (ANFIS). After a background presentation to each technology, the Modi Operandi (MO) for the digital fingerprints of cyberbullying and low-level drug dealing use cases were elaborated and the methodology was adapted anew, so as to fit the requirements of an evaluation with the two aforementioned technologies.

Chapter 5 presented the evaluation results for all the use cases examined in Chapter 3 and Chapter 4. Moreover, it provided results on unknown data testing for the use cases of Chapter 4. It was proved that all the techniques used did not show significant differences in performance and were able to perform pattern identification without significant issues for the nature of the problems they were given as inputs. However, the plain backpropagation NN showed the highest amount of advantages in terms of both performance, configuration and capability to solve complex problems.

6.2 Contributions

The main contributions of this thesis are as follows.

Contribution 1 - A Methodology for Suspicious Pattern Identification
Based on an extensive research on the relevant Mobile Forensic Data Analysis (MFDA) literature, we propose a new digital criminal profiling and suspicious pattern identification methodology that retains the evolutionary characteristics of its predecessors, such as the continuous interaction between the profiling characteristics and the new input data, but is also capable of assigning suspiciousness values to different data and metadata patterns.

Contribution 2 - Suspicious Pattern Identification with Fuzzy Systems
A proof of concept for the proposed methodology for a small use case scenario, aiming to profile the MO of PPDR officers defecting to the rioters' side by examining their sent SMS and to identify the respective suspicious patterns. Mamdani Fuzzy Systems with different configurations are used for the identification procedure and their performance is evaluated.

Contribution 3 - Suspicious Pattern Identification with NNs and ANFIS
Two complex use case scenarios, involving the profiling and identification of call and SMS patterns for cyberbullies and low-lever drug dealers are examined with the proposed methodology. NNs and ANFIS are configured and employed as evaluation tools.The performance of different setup is then measured, the prevailing solution is selected and tested anew on previously unknown data for the system.

The next section discusses the findings of the current thesis and proposes future research directions.

6.3 Discussion and Future Work

In this thesis, the performance of intelligent computation techniques for MF evidence analysis was evaluated. After creating a framework that incorporated the role of MF in a PPDR ecosystem, criminal investigation based on human behaviour of mobile device users in its internal and external environment was thoroughly researched.

A scenario of agent infiltration was created and a sample of SMS data consisting of three inputs was examined with the use of Mamdani Fuzzy Systems. Different fuzzy membership function configurations were used and the final suspiciousness results were calculated. They were then compared to the ground truth and their performance metrics were presented anew. The results from the procedure were satisfactory and the claim that Fuzzy Systems can be used as a means of pattern identification were verified.

However, as already mentioned in Chapter 4, Fuzzy Systems do not show the same level of efficiency when the input space consists of a relatively high number of inputs. Moreover, they have to be reconfigured for any alteration in the input and output data parameter ranges. Such a shortcoming renders them less useful for complex scenarios.

NNs and ANFIS were used for the investigation of two scenarios concerning human behaviour in the external environment of a PPDR system, in other words traditional criminal investigation. Call and SMS logs were pre-processed and examined for cyberbullying and low-level drug dealing use cases with a larger input space. Two types of NNs, a plain backpropagation perceptron and a pattern recognition backpropagation perceptron with thirteen different algorithms were tested for the NNs part. While both perceptrons showed excellent performance results, the plain backpropagation perceptron proved to be more stable in all the use case examples. Out of all the backpropagation algorithms, Levenberg-Marquardt and Bayesian Regularization backpropagation have proven to be more efficient.

ANFIS with different fuzzy clustering settings showed a small performance metric declining from the plain backpropagation perceptron, but still performed in a satisfactory manner. Its setup and configuration however are more complicated and time consuming than the almost plug-and-play equivalents of NNs. Nevertheless, they can still be used as an alternative methodology without significant losses in performance.

Lastly, the plain backpropagation configuration with was tested over a previously unknown dataset, deriving from a device that was used for experimental purposes. ADAET, a Python script responsible for the acquisition, pre-processing and evaluation bundle was developed and used with the test device. The produced results were almost excellent, with a declining of less than 10% percent from the original training and testing dataset. The Levenberg-Marquardt backpropagation algorithm was proven the more efficient anew.

NNs and ANFIS have proven that they can be used for pattern identification purposes in mobile forensic evidence. However, there are some key points that require improvement and can be considered as the current thesis‘future work.

- Use of data from official sources, such as law enforcement agencies or more detailed use-case simulations with actual devices can be used so as to provide a more stable testing background.

- Call and SMS logs are only a small sample of the data within a mobile device. New use cases, concerning other data sources, such as SNs logs, geolocation data, network usage and multimedia can be used so as to expand the boundaries of the current research.

- The experiments should not be limited strictly to handset evidence. Contemporary wearable devices are sources of valuable data that can serve as indicators for criminal investigation, such as heartbeat measurements and pedometers.

Lastly, an immediate future work concept is the expansion of the experimental phase of the current thesis with more use cases concerning traditional crimes and their subsequent evaluation, in order to observe and verify the existing scientific claims about performance and efficiency.

Bibliography

[504ensics Labs, 2013] 504ensics Labs (2013). Lime – linux memory extractor.

[Abraham, 2005] Abraham, A. (2005). Rule-based expert systems. *Handbook of measuring system design.*

[Al Barghouthy et al., 2013] Al Barghouthy, N. B., Marrington, A., and Baggili, I. (2013). The forensic investigation of android private browsing sessions using orweb. In *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*, pages 33–37.

[Al Barghouthy and Said, 2013] Al Barghouthy, N. B. and Said, H. (2013). Social networks im forensics: Encryption analysis. *Journal of Communications*, 8(11).

[Al-Saleh and Forihat, 2013] Al-Saleh, M. I. and Forihat, Y. A. (2013). Skype forensics in android devices. *International Journal of Computer Applications*, 78(7):38–44.

[Anglano, 2014] Anglano, C. (2014). Forensic analysis of whatsapp messenger on android smartphones. *Digital Investigation*, 11(3):201 – 213. Special Issue: Embedded Forensics.

[Ardagna et al., 2015] Ardagna, C. A., Asal, R., Damiani, E., and Vu, Q. H. (2015). From security to assurance in the cloud: A survey. *ACM Comput. Surv.*, 48(1):2:1–2:50.

[Augasta and Kathirvalavakumar, 2013] Augasta, M. G. and Kathirvalavakumar, T. (2013). Pruning algorithms of neural networks—a comparative study. *Central European Journal of Computer Science*, 3(3):105–115.

[Autopsy, 2016] Autopsy (2016). Autopsy - the sleuth kit.

[Ayers et al., 2014] Ayers, R., Brothers, S., and Jansen, W. (2014). Nist special publication 800-101, guidelines on mobile device forensics: Revision 1. Technical Report SP 800-101, National Institute of Standards and Technology.

[Badger et al., 2012] Badger, L., Grance, T., Patt-Corner, R., and Voas, J. (2012). Nist special publication 800-146, cloud computing synopsis and recommendations. Technical Report SP 800-146, National Institute of Standards and Technology.

[Baggili et al., 2015] Baggili, I., Oduro, J., Anthony, K., Breitinger, F., and McGee, G. (2015). Watch what you wear: Preliminary forensic analysis of smart watches. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 303–311.

[Baghirli, 2015] Baghirli, O. (2015). Comparison of lavenberg-marquardt, scaled conjugate gradient and bayesian regularization backpropagation algorithms for multistep ahead wind speed forecasting using multilayer perceptron feedforward neural network.

[Barbatsalou et al., 2015] Barbatsalou, K., Sousa, B., Monteiro, E., and Simoes, P. (2015). Mobile forensics for ppdr communications: How and why. page 30. Academic Conferences Limited.

[Barmpatsalou et al., 2017] Barmpatsalou, K., Cruz, T., Monteiro, E., and Simoes, P. (2017). Fuzzy system-based suspicious pattern detection in mobile forensic evidence. In *The 9th EAI International Conference on Digital Forensics and Cyber Crime*. EAI, European Alliance for Innovation.

[Barmpatsalou et al., 2018a] Barmpatsalou, K., Cruz, T., Monteiro, E., and Simoes, P. (2018a). Current and future trends in mobile forensics: A survey. *ACM Computing Surveys*, 51(3):1–31.

[Barmpatsalou et al., 2018b] Barmpatsalou, K., Cruz, T., Monteiro, E., and Simoes, P. (2018b). Fuzzy system-based suspicious pattern detection in mobile forensic evidence. In Matoušek, P. and Schmiedecker, M., editors, *Digital Forensics and Cyber Crime*, pages 106–114, Cham. Springer International Publishing.

[Barmpatsalou et al., 2013] Barmpatsalou, K., Damopoulos, D., Kambourakis, G., and Katos, V. (2013). A critical review of 7 years of mobile device forensics. *Digital Investigation*, 10(4):323 – 349.

[Battiti, 1992] Battiti, R. (1992). First-and second-order methods for learning: between steepest descent and newton's method. *Neural computation*, 4(2):141–166.

[Berenji and Khedkar, 1992] Berenji, H. R. and Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 3(5):724–740.

[Blej and Azizi, 2016] Blej, M. and Azizi, M. (2016). Comparison of mamdani-type and sugeno-type fuzzy inference systems for fuzzy real time scheduling. *International Journal of Applied Engineering Research*, 11(22):11071–11075.

[Borg and Groenen, 2005] Borg, I. and Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer.

[Borges et al., 2017] Borges, P., Sousa, B., Ferreira, L., Saghezchi, F. B., Mantas, G., Ribeiro, J., Rodriguez, J., Cordeiro, L., and Simoes, P. (2017). Towards a hybrid intrusion detection system for android-based ppdr terminals. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 1034–1039.

[Brownlee, 2017] Brownlee, J. (2017). What is the difference between test and validation datasets?

[Bui et al., 2012] Bui, D. T., Pradhan, B., Lofman, O., Revhaug, I., and Dick, O. B. (2012). Landslide susceptibility assessment in the hoa binh province of vietnam: A comparison of the levenberg–marquardt and bayesian regularized neural networks. *Geomorphology*, 171-172:12 – 29.

[Burden and Winkler, 2009] Burden, F. and Winkler, D. (2009). *Bayesian Regularization of Neural Networks*, pages 23–42. Humana Press, Totowa, NJ.

[Cahyani et al., 2016a] Cahyani, N. D. W., Martini, B., Choo, K.-K. R., and Al-Azhar, A. M. N. (2016a). Forensic data acquisition from cloud-of-things devices: Windows smartphones as a case study. *Concurrency and Computation: Practice and Experience*. CPE-16-0086.R1.

[Cahyani et al., 2016b] Cahyani, N. D. W., Rahman, N. H. A., Xu, Z., Glisson, W. B., and Choo, K.-K. R. (2016b). The role of mobile forensics in terrorism investigations involving the use of cloud apps. In *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*, MobiMedia '16, pages 199–204, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Casey, 2011] Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic Press, 3rd edition.

[Casey, 2013] Casey, E. (2013). Smartphone forensics and mobile malware analysis.

[CDMA Software, 2018] CDMA Software (2018). Cdma workshop.

[Cellebrite, 2018] Cellebrite (2018). Ufed ultimate.

[Cellebrite Predictions, 2015] Cellebrite Predictions (2015). Mobile forensics: A look ahead.

[Chabaa et al., 2010] Chabaa, S., Zeroual, A., and Antari, J. (2010). Identification and prediction of internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems and Applications*, 2(03):147.

[Chen et al., 2011] Chen, S.-W., Yang, C.-H., and Liu, C.-T. (2011). Design and implementation of live sd acquisition tool in android smart phone. In *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on*, pages 157–162.

[Cheng, 2011] Cheng, Y. (2011). Cybercrime forensic system in cloud computing. In *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, pages 612–615.

[Chung et al., 2012] Chung, H., Park, J., Lee, S., and Kang, C. (2012). Digital forensic investigation of cloud storage services. *Digital Investigation*, 9(2):81 – 95.

[D' Orazio et al., 2014] D' Orazio, C., Ariffin, A., and Choo, K.-K. R. (2014). ios anti-forensics: How can we securely conceal, delete and insert data? In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 4838–4847.

[D. T. Sacco and Tallon, 2010] D. T. Sacco, R. Argudin, J. M. and Tallon, K. (2010). Sexting: Youth practices and legal implications. *Cyberlaw Clinic.*

[da Silva et al., 2017] da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., and dos Reis Alves, S. F. (2017). Artificial neural network architectures and training processes. In *Artificial Neural Networks*, pages 21–28. Springer.

[Daryabar et al., 2015] Daryabar, F., Dehghantanha, A., and Choo, K.-K. R. (2015). Cloud storage forensics: Mega as a case study. *Australian Journal of Forensic Sciences*, 0(0):1–14.

[Demuth et al., 2014] Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. (2014). *Neural network design.* Martin Hagan.

[Dezfouli et al., 2015] Dezfouli, F. N., Dehghantanha, A., Eterovic-Soric, B., and Choo, K.-K. R. (2015). Investigating social networking applications on smartphones detecting facebook, twitter, linkedin and google+ artefacts on android and ios platforms. *Australian Journal of Forensic Sciences*, 0(0):1–20.

[Dezfouli et al., 2012] Dezfouli, F. N., Dehghantanha, A., Mahmoud, R., Sani, N. F. B. M., and bin Shamsuddin, S. (2012). Volatile memory acquisition using backup for forensic investigation. In *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, pages 186–189.

[Di Cerbo et al., 2011] Di Cerbo, F., Girardello, A., Michahelles, F., and Voronkova, S. (2011). Detection of malicious applications on android os. In Sako, H., Franke, K. Y., and Saitoh, S., editors, *Computational Forensics*, volume 6540 of *Lecture Notes in Computer Science*, pages 138–149. Springer Berlin Heidelberg.

[Digital Corpora, 2017] Digital Corpora (2017). Digital corpora: Producing the digital body.

[Do et al., 2015] Do, Q., Martini, B., and Choo, K.-K. R. (2015). A forensically sound adversary model for mobile devices. *PLoS ONE*, 10:e0138449.

[Edwards, 2013] Edwards, A. (2013). Drug dealer used unemployment benefits to make 24,000 mobile phone calls to sell heroin and crack.

[ELNEC, 2018] ELNEC (2018). Beeprog2.

[Enache et al., 2010] Enache, M., Hulea, M., and Leţia, T. S. (2010). Training neural networks for construction of informatics offender profile. In *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, volume 3, pages 1–6.

[Farina et al., 2015] Farina, J., Scanlon, M., Le-Khac, N.-A., and Kechadi, M. T. (2015). Overview of the forensic investigation of cloud services. In *Availability, Reliability and Security (ARES), 2015 10th International Conference on*, pages 556–565.

[Ferrari et al., 2008] Ferrari, S., Baumgartner, K. C., Palermo, G. B., Bruzzone, R., and Strano, M. (2008). Network models of criminal behavior. *IEEE Control Systems*, 28(4):65–77.

[Fielding, 2000] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis. AAI9980887.

[Figueiredo and Gomide, 1999] Figueiredo, M. and Gomide, F. (1999). Design of fuzzy systems using neurofuzzy networks. *IEEE Transactions on Neural Networks*, 10(4):815–827.

[Fleetwood, 2014] Fleetwood, J. (2014). Keeping out of trouble: Female crack cocaine dealers in england. *European Journal of Criminology*, 11(1):91–109.

[Fletcher, 1987] Fletcher, R. (1987). Practical methods of optimization john wiley & sons. *New York*, 80.

[Foresee and Hagan, 1997] Foresee, F. D. and Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Neural Networks,1997., International Conference on*, volume 3, pages 1930–1935 vol.3.

[Frontline Solvers, 2017] Frontline Solvers (2017). Training an artificial neural network.

[Fullér, 1995] Fullér, R. (1995). Neural fuzzy systems.

[Gacto et al., 2011] Gacto, M., Alcala, R., and Herrera, F. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340 – 4360. Special Issue on Interpretable Fuzzy Systems.

[Garcia and Zhou, 2010] Garcia, J. and Zhou, C. (2010). Improving gps precision and processing time using parallel and reduced-length wiener filters. *arXiv preprint arXiv:1006.0844*.

[Gegov, 2010] Gegov, A. (2010). *Fuzzy Networks for Complex Systems*. Springer.

[Ghosh and Chakraborty, 2012] Ghosh, A. and Chakraborty, M. (2012). Hybrid optimized back propagation learning algorithm for multi-layer perceptron. *International Journal of Computer Applications*, 60(13).

[Godwin, 2012] Godwin, M. (2012). Brief discussion on inductive/deductive profiling.

[Gomez-Miralles and Arnedo-Moreno, 2012] Gomez-Miralles, L. and Arnedo-Moreno, J. (2012). Versatile ipad forensic acquisition using the apple camera connection kit. *Computers & Mathematics with Applications*, 63(2):544 – 553. Advances in context, cognitive, and secure computing.

[Goodboy and Martin, 2015] Goodboy, A. K. and Martin, M. M. (2015). The personality profile of a cyberbully: Examining the dark triad. *Computers in Human Behavior*, 49:1 – 4.

[Gorzig and Olafsson, 2013] Gorzig, A. and Olafsson, K. (2013). What makes a bully a cyberbully? unravelling the characteristics of cyberbullies across twenty-five european countries. *Journal of Children and Media*, 7(1):9–27.

[Grajeda et al., 2017] Grajeda, C., Breitinger, F., and Baggili, I. (2017). Availability of datasets for digital forensics – and what is missing. *Digital Investigation*, 22:S94 – S105.

[Grispos et al., 2013] Grispos, G., Glisson, W. B., and Storer, T. (2013). Using smartphones as a proxy for forensic evidence contained in cloud storage services. *CoRR*, abs/1303.4078.

[Grispos et al., 2015] Grispos, G., Glisson, W. B., and Storer, T. (2015). Chapter 16 - recovering residual forensic data from smartphone interactions with cloud storage providers. In Choo, R. K.-K. R., editor, *The Cloud Security Ecosystem*, pages 347 – 382. Syngress, Boston.

[Grispos et al., 2011] Grispos, G., Storer, T., and Glisson, W. B. (2011). A comparison of forensic evidence recovery techniques for a windows mobile smart phone. *Digital Investigation*, 8(1):23–36.

[Grispos et al., 2012] Grispos, G., Storer, T., and Glisson, W. B. (2012). Calm before the storm: The challenges of cloud computing in digital forensics. *CoRR*, abs/1410.2123.

[Grover, 2013] Grover, J. (2013). Android forensics: Automated data collection and reporting from a mobile device. *Digital Investigation*, 10, Supplement(0):S12 – S20. The Proceedings of the Thirteenth Annual {DFRWS} Conference 13th Annual Digital Forensics Research Conference.

[Guidance Software, 2018] Guidance Software (2018). Encase forensic.

[Guillaume and Charnomordic, 2012] Guillaume, S. and Charnomordic, B. (2012). Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using fispro. *Expert Systems with Applications*, 39(10):8744 – 8755.

[Hanaysha et al., 2014] Hanaysha, T., Lindskog, D., and Ruhl, R. (2014). Using open source tools to investigate malware in the android operating system. In *Master of Information Systems Security Research 2014 Convocation, Concordia - University College of Alberta*, pages 1–8.

[Heaton, 2008] Heaton, J. (2008). *Introduction to Neural Networks with Java.* Heaton Research Inc., 2nd edition.

[Heriyanto, 2013] Heriyanto, A. P. (2013). Procedures and tools for acquisition and analysis of volatile memory on android smartphones.

[Hilgers et al., 2014] Hilgers, C., Macht, H., Müller, T., and Spreitzenbarth, M. (2014). Post-mortem memory analysis of cold-booted android devices. In *IT Security Incident Management IT Forensics (IMF), 2014 Eighth International Conference on*, pages 62–75.

[Ho et al., 2018] Ho, S. M., Kao, D., and Wu, W.-Y. (2018). Following the breadcrumbs: Timestamp pattern identification for cloud forensics. *Digital Investigation*, 24:79 – 94.

[Hoog, 2011] Hoog, A. (2011). *Android Forensics: Investigation, Analysis and Mobile Security for Google Android*. Syngress Publishing, 1st edition.

[Houmansadr et al., 2011] Houmansadr, A., Zonouz, S. A., and Berthier, R. (2011). A cloud-based intrusion detection and response system for mobile phones. In *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, pages 31–32.

[Iglewicz, 1983] Iglewicz, B. (1983). Robust scale estimators and confidence intervals forlocation. Wiley.

[Immanuel et al., 2015] Immanuel, F., Martini, B., and Choo, K.-K. R. (2015). Android cache taxonomy and forensic process. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 1094–1101.

[Islam and Verma, 2012] Islam, M. and Verma, V. K. (2012). Fuzzy logic based risk model for sms threats in 3g systems. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(2).

[J. T. Fish, L. S. Miller, M. C. Braswell, and E. W. Wallace Jr., 2014] J. T. Fish, L. S. Miller, M. C. Braswell, and E. W. Wallace Jr. (2014). The electronic crime scene. In *Crime Scene Investigation*. Anderson Publishing Ltd., Boston, 3rd edition.

[Jamieson, 2004] Jamieson, A. R. (2004). Radiocommunication for public protection and disaster relief. Technical report, International Telecommunication Union.

[Jang, 1993] Jang, J. S. R. (1993). Anfis: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

[Jang et al., 2015] Jang, J. W., Kang, H., Woo, J., Mohaisen, A., and Kim, H. K. (2015). Andro-autopsy: Anti-malware system based on similarity matching of malware and malware creator-centric information. *Digital Investigation*, 14:17 – 35.

[Jansen and Ayers, 2007] Jansen, W. and Ayers, R. P. (2007). Nist special publication 800-101, guidelines on cell phone forensics. Technical Report SP 800-101, Gaithersburg, MD, United States.

[Jin, 2012] Jin, Y. (2012). *Advanced fuzzy systems design and applications*, volume 112. Physica.

[Juang and Lin, 1998] Juang, C.-F. and Lin, C.-T. (1998). An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1):12–32.

[Kaart and Laraghy, 2014] Kaart, M. and Laraghy, S. (2014). Android forensics: Interpretation of timestamps. *Digital Investigation*, 11(3):234 – 248. Special Issue: Embedded Forensics.

[Kar et al., 2014] Kar, S., Das, S., and Ghosh, P. K. (2014). Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing*, 15:243–259.

[Kasabov and Song, 2002] Kasabov, N. K. and Song, Q. (2002). Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *Trans. Fuz Sys.*, 10(2):144–154.

[Kasiaras et al., 2014] Kasiaras, D., Zafeiropoulos, T., Clarke, N., and Kambourakis, G. (2014). Android forensics: Correlation analysis. In *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for*, pages 157–162.

[Kechadi et al., 2015] Kechadi, M. T., Faheem, M., and Le-Khac, N. A. (2015). The state of the art forensic techniques in mobile cloud environment: A survey, challenges and current trends. *Int. J. Digit. Crime For.*, 7(2):1–19.

[Khan et al., 2014] Khan, A. u. R., Othman, M., Madani, S. A., and Khan, S. U. (2014). A survey of mobile cloud computing application models. *Communications Surveys Tutorials, IEEE*, 16(1):393–413.

[Klaver, 2010] Klaver, C. (2010). Windows mobile advanced forensics. *Digital Investigation*, 6(3-4):147–167.

[Kocsis, 2006] Kocsis, R. N. (2006). *What Is Criminal Profiling?* Springer.

[Kohn et al., 2013] Kohn, M. D., Eloff, M. M., and Eloff, J. H. (2013). Integrated digital forensic process model. *Computers & Security*, 38(0):103 – 115. Cybercrime in the Digital Economy.

[Kohonen, 1982] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69.

[Kotsopoulos and Stamatiou, 2012] Kotsopoulos, P. A. and Stamatiou, Y. (2012). Uncovering mobile phone users' malicious activities using open source tools. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 927–933.

[Kumar and Verma, 2015] Kumar, D. and Verma, P. (2015). Comparative study of mamdani takagi-sugeno models for spectrum access in cognitive radio networks. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–5.

[Kuruvilla and Gunavathi, 2014] Kuruvilla, J. and Gunavathi, K. (2014). Lung cancer classification using neural networks for ct images. *Computer Methods and Programs in Biomedicine*, 113(1):202 – 209.

[Kwan et al., 2008] Kwan, L., Ray, P., and Stephens, G. (2008). Towards a methodology for profiling cyber criminals. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 264–264. IEEE.

[Lai et al., 2013] Lai, P., Chow, K.-P., Fan, X.-X., and Chan, V. (2013). An empirical study profiling internet pirates. In Peterson, G. and Shenoi, S., editors, *Advances in Digital Forensics IX*, pages 257–272, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Lee and Hong, 2011] Lee, J. and Hong, D. W. (2011). Pervasive forensic analysis based on mobile cloud computing. pages 572 – 576, Shanghai. IEEE.

[Li et al., 2012] Li, J., Gu, D., and Luo, Y. (2012). Android malware forensics: Reconstruction of malicious events. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 552–558.

[Lievens, 2014] Lievens, E. (2014). Bullying and sexting in social networks: Protecting minors from criminal acts or empowering minors to cope with risky behaviour? *International Journal of Law, Crime and Justice*, 42(3):251 – 270.

[Lilly, 2010] Lilly, J. H. (2010). *Fuzzy Control and Identification (Lilly/Fuzzy Control) || Mamdani Fuzzy Systems.*

[Lima and Camargo, 2014] Lima, H. P. D. and Camargo, H. d. A. (2014). A methodology for building fuzzy rule-based systems integrating expert and data knowledge. In *2014 Brazilian Conference on Intelligent Systems*, pages 300–305.

[Lin and Lee, 1991] Lin, C.-T. and Lee, C. S. G. (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on computers*, 40(12):1320–1336.

[Luttenberger and Creutzburg, 2011] Luttenberger, S. and Creutzburg, R. (2011). Forensic investigation of certain types of mobile devices. volume 7881, pages 78810Q–78810Q–12.

[Mamdani, 1974] Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Electrical Engineers, Proceedings of the Institution of*, 121(12):1585–1588.

[Martini and Choo, 2012] Martini, B. and Choo, K.-K. R. (2012). An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation*, 9(2):71 – 80.

[Martini and Choo, 2014] Martini, B. and Choo, K.-K. R. (2014). Cloud forensic technical challenges and solutions: A snapshot. *Cloud Computing, IEEE*, 1(4):20–25.

[Martini et al., 2015a] Martini, B., Do, Q., and Choo, K.-K. R. (2015a). Chapter 14 - conceptual evidence collection and analysis methodology for android devices. In Choo, R. K.-K. R., editor, *The Cloud Security Ecosystem*, pages 285 – 307. Syngress, Boston.

[Martini et al., 2015b] Martini, B., Do, Q., and Choo, K.-K. R. (2015b). Chapter 15 - mobile cloud forensics: An analysis of seven popular android apps. In Choo, R. K.-K. R., editor, *The Cloud Security Ecosystem*, pages 309 – 345. Syngress, Boston.

[Marturana et al., 2011] Marturana, F., Me, G., Berte, R., and Tacconi, S. (2011). A quantitative approach to triaging in mobile forensics. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 582–588.

[Marturana et al., 2012] Marturana, F., Me, G., and Tacconi, S. (2012). A case study on digital forensics in the cloud. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*, pages 111–116.

[May and Hough, 2004] May, T. and Hough, M. (2004). Drug markets and distribution systems. *Addiction Research & Theory*, 12(6):549–563.

[McEwen, 2011] McEwen, R. N. (2011). Tools of the trade: Drugs, law and mobile phones in canada. *New Media & Society*, 13(1):134–150.

[Mell and Grance, 2011] Mell, P. and Grance, T. (2011). The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD.

[Mena, 2003] Mena, J. (2003). *Investigative Data Mining for Security and Criminal Detection*. Butterworth–Heinemann.

[Michalas and Murray, 2016] Michalas, A. and Murray, R. (2016). Mem tri: Memory forensics triage tool. Technical report, Cyber Security Group, University of Westminster.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

[Mohanty et al., 2010] Mohanty, S., Jha, M. K., Kumar, A., and Sudheer, K. P. (2010). Artificial neural network modeling for groundwater level forecasting in a river island of eastern india. *Water Resources Management*, 24(9):1845–1865.

[Mokhonoana and Olivier, 2007] Mokhonoana, P. M. and Olivier, M. S. (2007). Acquisition of a symbian smart phone's content with an on-phone forensic tool. In *Southern African telecommunication networks and applications conference*.

[Motahari-Nezhad et al., 2009] Motahari-Nezhad, H. R., Stephenson, B., and Singhal, S. (2009). Outsourcing business to cloud computing services: Opportunities and challenges. *IEEE Internet Computing, Special Issue on Cloud Computing*, 10.

[Müller and Spreitzenbarth, 2013] Müller, T. and Spreitzenbarth, M. (2013). Frost: Forensic recovery of scrambled telephones. In *Proceedings of the 11th International Conference on Applied Cryptography and Network Security*, ACNS'13, pages 373–388, Berlin, Heidelberg. Springer-Verlag.

[Murphy, 2013] Murphy, C. A. (2013). Developing process for mobile device forensics.

[Mutawa et al., 2016] Mutawa, N. A., Bryce, J., Franqueira, V. N., and Marrington, A. (2016). Forensic investigation of cyberstalking cases using behavioural evidence analysis. *Digital Investigation*, 16:S96 – S103. DFRWS 2016 Europe.

[Nassif and Hruschka, 2011] Nassif, L. F. d. C. and Hruschka, E. R. (2011). Document clustering for forensic computing: An approach for improving computer inspection. In *Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops - Volume 01*, ICMLA '11, pages 265–268, Washington, DC, USA. IEEE Computer Society.

[Natarajan, 2006] Natarajan, M. (2006). Understanding the structure of a large heroin distribution network: A quantitative analysis of qualitative data. *Journal of Quantitative Criminology*, 22(2):171–192.

[National Institute of Standards and Technology (NIST), 2016] National Institute of Standards and Technology (NIST) (2016). The cfreds project.

[Nauck and Kruse, 1994] Nauck, D. and Kruse, R. (1994). Nefcon-i: An x-window based simulator for neural fuzzy controllers. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, volume 3, pages 1638–1643. IEEE.

[Netherlands Forensic Institute, 2018] Netherlands Forensic Institute (2018). Nfi memory toolkit.

[Nguli et al., 2014] Nguli, D., Graziano, A., Nicolaou, G., and Fredrick, J. (2014). Nyuki android process dumper user guide.

[Ninawe and Ardhapurkar, 2014] Ninawe, P. N. and Ardhapurkar, S. B. (2014). Forensic-as-a-service for mobile devices (literature survey). *(IJCSIT) International Journal of Computer Science and Information Technologies*, 5(6):7776–7778.

[NowSecure, 2016] NowSecure (2016). Nowsecure: Power. efficient mf for android and ios.

[Ntantogian et al., 2014] Ntantogian, C., Apostolopoulos, D., Marinakis, G., and Xenakis, C. (2014). Evaluating the privacy of android mobile applications under forensic analysis. *Computers & Security*, 42(0):66 – 76.

[Openet, 2013] Openet (2013). Survey: 41% of teens experience cyber-bullying on their cellies.

[Palmer, 2001] Palmer, G. (2001). A road map for digital forensic research. Technical Report DTRT0010-01, Digital Forensic Research Workshop (DFRWS).

[Panchal et al., 2011] Panchal, G., Ganatra, A., Kosta, Y., and Panchal, D. (2011). Behaviour analysis of multilayer perceptronswith multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2):332.

[Pande, 2013] Pande, G. (2013). Performance evaluation of video communications over 4g network. *CoRR*, abs/1305.1887.

[Pang, 2015] Pang, A. (2015). Worldwide cloud applications market forecast 2015-2019.

[Pilli et al., 2010] Pilli, E. S., Joshi, R. C., and Niyogi, R. (2010). Network forensic frameworks: Survey and research challenges. *Digital Investigation*, 7(1–2):14 – 27.

[Platzer et al., 2014] Platzer, C., Stuetz, M., and Lindorfer, M. (2014). Skin sheriff: A machine learning solution for detecting explicit images. In *Proceedings of the 2Nd International Workshop on Security and Forensics in Communication Systems*, SFCS '14, pages 45–56, New York, NY, USA. ACM.

[Quick and Choo, 2017] Quick, D. and Choo, K.-K. R. (2017). Pervasive social networking forensics: Intelligence and evidence from mobile device extracts. *Journal of Network and Computer Applications*, 86:24 – 33. Special Issue on Pervasive Social Networking.

[Raghav and Saxena, 2009] Raghav, S. and Saxena, A. K. (2009). Mobile forensics: Guidelines and challenges in data preservation and acquisition. In *Research and Development (SCOReD), 2009 IEEE Student Conference on*, pages 5–8.

[Riedmiller, 1994] Riedmiller, M. (1994). Advanced supervised learning in multi-layer perceptrons — from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, 16(3):265 – 278.

[RightScale, 2016] RightScale (2016). State of the cloud report.

[Roberto et al., 2014] Roberto, A. J., Eden, J., Savage, M. W., Ramos-Salazar, L., and Deiss, D. M. (2014). Prevalence and predictors of cyberbullying perpetration by high school seniors. *Communication Quarterly*, 62(1):97–114.

[Rogers, 2003] Rogers, M. (2003). The role of criminal profiling in the computer forensics process. *Computers & Security*, 22(4):292 – 298.

[Rogers, 2016] Rogers, M. K. (2016). *Digital Forensics: Threatscape and Best Practices*. Elsevier, 1st edition.

[Rogers et al., 2006] Rogers, M. K., Goldman, J., Mislan, R., Wedge, T., and Debrota, S. (2006). Computer forensics field triage process model. *Journal of Digital Forensics, Security and Law*, 1(2):19–38.

[Ruan et al., 2013] Ruan, K., Carthy, J., Kechadi, M. T., and Baggili, I. (2013). Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. *Digital Investigation*, 10(1):34 – 43.

[Saikia et al., 2011] Saikia, L. C., Mishra, S., Sinha, N., and Nanda, J. (2011). Automatic generation control of a multi area hydrothermal system using reinforced learning neural network controller. *International Journal of Electrical Power & Energy Systems*, 33(4):1101–1108.

[Saltaformaggio et al., 2015] Saltaformaggio, B., Bhatia, R., Gu, Z., Zhang, X., and Xu, D. (2015). Guitar: Piecing together android app guis from memory images. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 120–132, New York, NY, USA. ACM.

[SALUS, 2014] SALUS (2014). Deliverable 7.1 salus ppdr platform - intermediate.

[Samet et al., 2014] Samet, N., Letaifa, A. B., Hamdi, M., and Tabbane, S. (2014). Forensic investigation in mobile cloud environment. In *Networks, Computers and Communications, The 2014 International Symposium on*, pages 1–5.

[Schutz et al., 2013] Schutz, P., Breuer, M., Hofken, H., and Schuba, M. (2013). Malware proof on mobile phone exhibits based on gsm/gprs traces. In *The Second International Conference on Cyber Security, Cyber Peacefare and Digital Forensic (CyberSec2013)*, pages 89–96.

[Shaharin et al., 2014] Shaharin, R., Prodhan, U. K., and Rahman, M. (2014). Performance study of tdnn training algorithm for speech recognition. *Int. J. Adv. Res. Comput. Sci. Technol*, 2:90–95.

[Shanableh, 2013] Shanableh, T. (2013). Detection of frame deletion for digital video forensics. *Digital Investigation*, 10(4):350 – 360.

[Shariati et al., 2016] Shariati, M., Dehghantanha, A., and Choo, K.-K. R. (2016). Sugarsync forensic analysis. *Australian Journal of Forensic Sciences*, 48(1):95–117.

[Shariati et al., 2015] Shariati, M., Dehghantanha, A., Martini, B., and Choo, K.-K. R. (2015). Chapter 19 - ubuntu one investigation: Detecting evidences on client machines. In Choo, R. K.-K. R., editor, *The Cloud Security Ecosystem*, pages 429 – 446. Syngress, Boston.

[Sheela and Deepa, 2013] Sheela, K. G. and Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, 2013.

[Siddique and Adeli, 2013] Siddique, N. and Adeli, H. (2013). *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley & Sons.

[Silde and Angelopoulou, 2014] Silde, A. and Angelopoulou, O. (2014). A digital forensics profiling methodology for the cyberstalker. In *2014 International Conference on Intelligent Networking and Collaborative Systems*, pages 445–450.

[Silensec, 2016] Silensec (2016). Nyuki forensic investigator (nfi).

[Singh et al., 2006] Singh, T. N., Gupta, A. R., and Sain, R. (2006). A comparative analysis of cognitive systems for the prediction of drillability of rocks and wear factor. *Geotechnical & Geological Engineering*, 24(2):299–312.

[Sivanandam and Deepa, 2006] Sivanandam, S. and Deepa, S. (2006). *Introduction to neural networks using Matlab 6.0*. Tata McGraw-Hill Education.

[Smith, 2008] Smith, P. K. (2008). Cyberbullying: its nature and impact in secondary school pupils. *J Child Psychol Psychiatry*, 49(4):376 – 385.

[Soft Center, 2018] Soft Center (2018). Flash extractor.

[Souppaya and Scarfone, 2013] Souppaya, M. and Scarfone, K. (2013). Nist special publication 800-124, guidelines for managing the security of mobile devices in the enterprise: Revision 1. Technical report, National Institute of Standards & Technology.

[Spreitzenbarth and Uhrmann, 2015] Spreitzenbarth, M. and Uhrmann, J. (2015). *Mastering Python Forensics*. Packt Publishing Ltd.

[Strano, 2004] Strano, M. (2004). A neural network applied to criminal psychological profiling: An italian initiative. *International Journal of Offender Therapy and Comparative Criminology*, 48(4):495–503.

[Sulzberger et al., 1993] Sulzberger, S. M., Tschichold-Gurman, N., and Vestli, S. J. (1993). Fun: Optimization of fuzzy rule based systems using neural networks. In *Neural Networks, 1993., IEEE International Conference on*, pages 312–316. IEEE.

[Suparta and Alhasa, 2016] Suparta, W. and Alhasa, K. M. (2016). Adaptive neuro-fuzzy interference system. In *Modeling of Tropospheric Delays Using ANFIS*, pages 5–18. Springer.

[Takagi and Sugeno, 1993] Takagi, T. and Sugeno, M. (1993). Fuzzy identification of systems and its applications to modeling and control. In *Readings in Fuzzy Sets for Intelligent Systems*, pages 387–403. Elsevier.

[Tano et al., 1996] Tano, S., Oyama, T., and Arnould, T. (1996). Deep combination of fuzzy inference and neural network in fuzzy inference software — finest. *Fuzzy Sets and Systems*, 82(2):151 – 160. Connectionist and Hybrid Connectionist Systems for Approximate Reasoning.

[Thing et al., 2010] Thing, V. L., Ng, K.-Y., and Chang, E.-C. (2010). Live memory forensics of mobile phones. *Digital Investigation*, 7, Supplement(0):S74 – S82. The Proceedings of the Tenth Annual {DFRWS} Conference.

[Thomson, 2012] Thomson, G. (2012). Byod: enabling the chaos. *Network Security*, 2012(2):5 – 8.

[Ticknor, 2013] Ticknor, J. L. (2013). A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, 40(14):5501 – 5506.

[Turvey, 2011] Turvey, B. E. (2011). *Criminal Profiling: An Introduction to Behavioral Evidence Analysis*. Elsevier.

[UNODC, 2011] UNODC (2011). *Criminal Intelligence: Manual for Analysts*. United Nations Publications.

[Vieira et al., 2004] Vieira, J., Dias, F. M., and Mota, A. (2004). Neuro-fuzzy systems: a survey. In *5th WSEAS NNA international conference on neural networks and applications, Udine, Italia*.

[Viharos and Kis, 2015] Viharos, Z. and Kis, K. (2015). Survey on neuro-fuzzy systems and their applications in technical diagnostics and measurement. *Measurement*, 67:126 – 136.

[Volatile Systems, 2011] Volatile Systems (2011). The volatility framework: volatile memory artifact extraction utility framework.

[Vomel, 2013] Vomel, S. (2013). *Forensic Acquisition and Analysis of Volatile Data in Memory*. PhD thesis, Fakult at der Friedrich-Alexander, Universitat Erlangen-Nurnberg.

[Votipka et al., 2013] Votipka, D., Vidas, T., and Christin, N. (2013). Passe-partout: A general collection methodology for android devices. *Information Forensics and Security, IEEE Transactions on*, 8(12):1937–1946.

[Wagner et al., 2014] Wagner, D. T., Rice, A., and Beresford, A. R. (2014). Device analyzer: Understanding smartphone usage. In Stojmenovic, I., Cheng, Z., and Guo, S., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 195–208, Cham. Springer International Publishing.

[Walls et al., 2011] Walls, R., Learned-Miller, E., and Levine, B. N. (2011). Forensic triage for mobile phones with dec0de. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 7–7, Berkeley, CA, USA. USENIX Association.

[Widrow and Hoff, 1960] Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. Technical report, STANFORD UNIV CA STANFORD ELECTRONICS LABS.

[Wilamowski and Yu, 2010] Wilamowski, B. M. and Yu, H. (2010). Improved computation for levenberg - marquardt training. *IEEE Transactions on Neural Networks*, 21(6):930–937.

[Xia et al., 2010] Xia, J. H., Rusli, and Kumta, A. S. (2010). Feedforward neural network trained by bfgs algorithm for modeling plasma etching of silicon carbide. *IEEE Transactions on Plasma Science*, 38(2):142–148.

[Yuan et al., 2003] Yuan, H., Xiong, F., and Huai, X. (2003). A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. *Computers and Electronics in Agriculture*, 40(1-3):57–64.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.

[Zadeh, 1973] Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28–44.

[Zawoad et al., 2013] Zawoad, S., Dutta, A. K., and Hasan, R. (2013). Seclaas: Secure logging-as-a-service for cloud forensics. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 219–230, New York, NY, USA. ACM.

[Zawoad and Hasan, 2013] Zawoad, S. and Hasan, R. (2013). Cloud forensics: A meta-study of challenges, approaches, and open problems. *CoRR*, abs/1302.6312.

[Zawoad and Hasan, 2015] Zawoad, S. and Hasan, R. (2015). Towards a systematic analysis of challenges and issues in secure mobile cloud forensics. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*, pages 237–238.

[Zdziarski, 2008] Zdziarski, J. (2008). ios forensic investigative methods. Technical report, International Telecommunication Union.

[Zimmermann, 1996] Zimmermann, H.-J. (1996). Fuzzy control. In *Fuzzy Set Theory—and Its Applications*, pages 203–240. Springer.