ON DESCRIPTIVE AND PREDICTIVE MODELS FOR SERIAL CRIME ANALYSIS

Anton Borg

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2014:12
Department of Computer Science and Engineering
On Descriptive and Predictive Models for Serial Crime Analysis

Anton Borg
On Descriptive and Predictive Models for Serial Crime Analysis

Anton Borg

Doctoral dissertation in Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN
A learning experience is one of those things that say, "You know that thing you just did? Don’t do that."

Abstract

Law enforcement agencies regularly collect crime scene information. There exists, however, no detailed, systematic procedure for this. The data collected is affected by the experience or current condition of law enforcement officers. Consequently, the data collected might differ vastly between crime scenes. This is especially problematic when investigating volume crimes.

Law enforcement officers regularly do manual comparison on crimes based on the collected data. This is a time-consuming process; especially as the collected crime scene information might not always be comparable. The structuring of data and introduction of automatic comparison systems could benefit the investigation process. This thesis investigates descriptive and predictive models for automatic comparison of crime scene data with the purpose of aiding law enforcement investigations.

The thesis first investigates predictive and descriptive methods, with a focus on data structuring, comparison, and evaluation of methods. The knowledge is then applied to the domain of crime scene analysis, with a focus on detecting serial residential burglaries. This thesis introduces a procedure for systematic collection of crime scene information. The thesis also investigates impact and relationship between crime scene characteristics and how to evaluate the descriptive model results.

The results suggest that the use of descriptive and predictive models can provide feedback for crime scene analysis that allows a more effective use of law enforcement resources. Using descriptive models based on crime characteristics, including Modus Operandi, allows law enforcement agents to filter cases intelligently. Further, by estimating the link probability between cases, law enforcement agents can focus on cases with higher link likelihood. This would allow a more effective use of law enforcement resources, potentially allowing an increase in clear-up rates.
Sammanfattning

Antalet bostadsinbrott som begås årligen i Sverige har ökat de senaste 10 åren. 2013 anmäldes ungefär 22 000 bostadsinbrott och av dessa löser polisen ungefär 3-5%. Enligt polisen begås ett stort antal av de anmälda brotten av så kallade mobila vinningskriminella. Det vill säga ligo som åker runt och begär inbrott i vinstdrivande syfte. Polisen kan knyta samman flera bostadsinbrott genom att hitta kopplingar mellan inbrott, t ex samma sorts stulet gods eller liknande ingångsmetod. På grund av mängden anmälda bostadsinbrott kan detta vara svårt.

Information från brottsplatser samlas regelbundet in av polisen. Dock saknas en systematisk metod för insamling. Varje polis avgör till viss del själv vilken information som är relevant att samla in från brottsplatsen. Detta medför att information som samlas in från olika brottsplatser skiljer sig åt, både i vilken typ av information som samlas in och i kvaliteten på den insamlade informationen. Detta försvårar senare jämförelser mellan brottsplatser, vilket är särskilt påtagligt när man undersöker exempelvis bostadsinbrott på grund av den stora mängden sådana brott.


Avhandling är uppdelad i två delar. I den första delen undersöks metoder för lärande system, med fokus på hur man strukturerar data, samt jämför och utvärderar resultaten. Metoderna som

Acknowledgments

I would like to thank my supervisors, Dr. Niklas Lavesson and Dr. Martin Boldt, for their never-ending support. Without them I could never have finished this work.

I would also like to thank Professor Bengt Carlsson in his role as examiner.

Professor Veselka Boeva, for getting me started with clustering approaches and pointing me in the right direction.

Finally, I would like to thank the Swedish Police, especially Detective Inspector Ulf Melander, and the Swedish National Laboratory of Forensic Science for their support and assistance with domain expertise and data collection in this work.

This work was partly founded by .SE, the Internet Infrastructure Foundation, and the European Regional Development Fund (ERDF).
This compilation thesis consists of six articles that have been peer reviewed and published in conference proceedings or journals, or submitted for publication. The articles have been authored by the thesis author or co-authored with senior colleagues. The following publications are included:


5. Anton Borg, "Linking Residential Burglaries", *Submitted for journal publication*.

6. Anton Borg, Martin Boldt, "Combining Modus Operandi for Clustering Burglaries", *Submitted for journal publication*. 
Authorship

Publication 1 extends previous research [1], adding automatic extraction and processing of End User License Agreements. For this publication, the thesis author was the main driver in the investigation. The involvement comprised in setting up the experiment design, writing the paper and analyzing the data. For publication 2, the thesis author was the main driver in the experiment design, writing the paper, analyzing data and designing the algorithm. For publication 3, the thesis author was the main driver in the experiment design, and analyzing the data. The thesis author was the main driver in writing the paper, but it was co-written with the third author. For publication 4, the thesis author was the main driver in the experiment design, writing the paper, analyzing data. For publication 5, the thesis author was the sole author. The paper replicates and extends previous research. For publication 6, the thesis author was the main driver in the experiment design, and analyzing data. The thesis author was the main driver in writing the paper, but it was co-written with the second author. The publication continues the work in paper 4.

Publication relationships

The thesis is divided into two parts. Part one concerns foundations of predictive and descriptive methodology. It consists of publication 1 through 3 and the lessons learned in these publication are used and applied in the publications in part two of the thesis. Part two consists of publication 4 through 6 and concerns how descriptive and predictive models can be used to aid in the investigation of series of residential burglaries.
Related papers

The following publications are related, but not included in the thesis.


Funding

Publication 1 and Publication 2 was funded by .SE, the Internet Infrastructure Foundation. Publication 4 through Publication 6 was funded by the European Regional Development Fund (ERDF).
Contents

Acknowledgments vii
Preface ix
Contents xiii
List of Figures xix
List of Tables xxi

1 Introduction 1
  1.1 Aim & Scope ........................................ 3
  1.2 Outline ........................................... 4

2 Background 5
  2.1 Terminology ....................................... 8
  2.2 Related Work ...................................... 10

3 Approach 13
  3.1 Research Questions ................................. 14
  3.2 Research Methodology ............................... 15
  3.3 Validity Threats .................................... 16

4 Results 19
  4.1 Contributions ..................................... 19
CONTENTS

4.2 Discussion ............................................. 23
4.3 Conclusion ........................................... 28
4.4 Future Work .......................................... 28

5 Informed Software Installation through License Agreement Categorization 31
Anton Borg, Martin Boldt and Niklas Lavesson
5.1 Introduction ........................................... 32
   5.1.1 Aim and Scope .................................. 32
   5.1.2 Outline .......................................... 33
5.2 Background and Related Work .......................... 33
   5.2.1 Background ...................................... 33
   5.2.2 Machine Learning ................................. 34
   5.2.3 Related Work .................................... 36
5.3 Approach .............................................. 37
   5.3.1 Automated System ................................. 38
   5.3.2 Data Preprocessing ................................. 41
5.4 Experimental procedure ................................ 43
   5.4.1 Experiment 1: Feature Selection .................. 44
   5.4.2 Experiment 2: Parameter Tuning .................. 44
   5.4.3 Evaluation Metrics ................................. 46
5.5 Results ................................................ 48
   5.5.1 Experiment 1 ..................................... 48
   5.5.2 Experiment 2 ..................................... 48
5.6 Discussion ............................................ 51
   5.6.1 Data Set Content ................................. 51
   5.6.2 Proposed System Vulnerabilities ................. 52
   5.6.3 Experimental Results ............................. 53
5.7 Conclusion and Future Work ........................... 54

6 Social Network-based E-mail Classification 57
Anton Borg, Niklas Lavesson
CONTENTS

6.1 Introduction .............................................. 58
   6.1.1 Aim and Scope ..................................... 58
   6.1.2 Outline ............................................. 58
6.2 Background ............................................. 59
6.3 Related Work ........................................... 60
6.4 Theoretical Model ..................................... 62
   6.4.1 Data Sources ....................................... 63
   6.4.2 Context-driven Classification ..................... 63
   6.4.3 Knowledge-based Classification ................... 64
   6.4.4 Automatic E-mail Classification .................. 64
6.5 Method ................................................ 65
   6.5.1 Social Data Generation ............................ 65
   6.5.2 Social Data Metrics ............................... 66
6.6 Experiments ............................................ 67
   6.6.1 Data Collection .................................... 67
   6.6.2 Data Preprocessing ................................ 67
   6.6.3 Feature Selection .................................. 69
   6.6.4 Algorithm Selection ............................... 69
   6.6.5 Performance Evaluation ............................ 71
6.7 Results ................................................ 71
6.8 Discussion ............................................. 72
   6.8.1 Social Network Information ....................... 74
6.9 Conclusion and Future Work ........................... 75

7 Comparison of Clustering Approaches for Gene Expression Data 77
   Anton Borg, Niklas Lavesson, Veselka Boeva
7.1 Introduction ............................................ 78
7.2 Related Work .......................................... 79
7.3 Methods ................................................. 80
   7.3.1 Cut-clustering Algorithm ......................... 80
   7.3.2 K-means Clustering Algorithm .................... 82
   7.3.3 Spectral Clustering Algorithm .................... 82
## CONTENTS

7.3.4 Expectation-maximisation Clustering Algorithm . . 82  
7.4 Experimental Setup ............................................. 83  
7.4.1 Microarray Datasets ........................................... 83  
7.4.2 Cluster Validation Measures ............................... 84  
7.5 Validation Results and Discussion ............................. 87  
7.5.1 Clustering Quality ............................................ 87  
7.5.2 Clustering Stability ........................................... 89  
7.5.3 Clustering Accuracy ........................................... 91  
7.6 Conclusion ....................................................... 92  

8 Detecting Serial Residential Burglaries using Clustering 93  
Anton Borg, Martin Boldt, Niklas Lavesson, Ulf Melander,  
Veselka Boeva  
8.1 Introduction ..................................................... 95  
8.1.1 Purpose Statement ........................................... 97  
8.1.2 Outline ....................................................... 97  
8.2 Decision Support System for Residential Burglary Analysis 97  
8.3 Related Work ................................................... 100  
8.4 Cut Clustering Algorithm ....................................... 102  
8.4.1 The α Value ................................................ 104  
8.4.2 Minimum Cut Tree .......................................... 104  
8.5 Data and Method ............................................... 105  
8.5.1 Data Collection ............................................. 105  
8.5.2 Data Representation ....................................... 107  
8.5.3 Cluster Validation Measurements .......................... 108  
8.6 Experiment Design ............................................. 110  
8.6.1 Hypothesis .................................................. 110  
8.6.2 Experiment 1: Cluster Quality ............................ 110  
8.6.3 Experiment 2: Crime Distinction .......................... 111  
8.7 Results ......................................................... 112  
8.7.1 Experiment 1 ................................................ 112  
8.7.2 Experiment 2 ................................................ 114  
8.8 Analysis ........................................................ 118
## CONTENTS

### 8.8.1 Experiment 1 ................................. 118
### 8.8.2 Experiment 2 ................................. 119
### 8.8.3 Validity Threats .............................. 123
### 8.8.4 Discussion ................................. 124
### 8.9 Conclusion ................................. 127

### 9 Linking Residential Burglaries ............................. 129

Anton Borg

*Submitted for publication.*

#### 9.1 Introduction ........................................ 130
  9.1.1 Aims and Scope .................................. 131
  9.1.2 Contributions .................................. 131

#### 9.2 Background ........................................ 131

#### 9.3 Related Work ....................................... 132

#### 9.4 Data ............................................. 134
  9.4.1 Data Collection .................................. 134
  9.4.2 Data Preparation .................................. 134

#### 9.5 Method and Experiment Setup ..................... 135
  9.5.1 Evaluation Metrics .............................. 135
  9.5.2 Evaluation ...................................... 136

#### 9.6 Residential Burglary Characteristics .................. 137

#### 9.7 Results ........................................ 138

#### 9.8 Discussion ....................................... 144

#### 9.9 Conclusion ....................................... 145

### 10 Combining Modus Operandi for Clustering Burglaries ............................. 147

Anton Borg, Martin Boldt

*Submitted for publication.*

#### 10.1 Introduction ...................................... 148
  10.1.1 Purpose Statement ............................. 148
  10.1.2 Outline ...................................... 149

#### 10.2 Related Work ..................................... 149

#### 10.3 Data ........................................... 150

#### 10.4 Method ......................................... 152
  10.4.1 Distance Metric .............................. 152
10.4.2 Clustering Algorithms ........................................ 154
10.4.3 Evaluation Metrics ............................................. 156
10.5 Results ................................................................. 158
10.6 Result Analysis ....................................................... 161
  10.6.1 Distance Metric Comparison ................................ 161
  10.6.2 Algorithm Comparison ........................................ 163
  10.6.3 Evaluation Metric Analysis ................................... 164
10.7 Discussion ............................................................. 166
10.8 Conclusion ............................................................ 168
10.9 Future work ............................................................ 169

References ........................................................................ 171

A Edge Representation and Removal Criteria ....................... 185
  A.1 Edge Representation ................................................. 185
  A.2 Edge Removal Criteria .............................................. 186

B Standardised Complaint Form ...................................... 191
List of Figures

1.1 Entry MO characteristics example ......................... 2
1.2 A view of local crimes for a specific search in the suggested DSS. .................................................. 3
2.1 Total number of reported burglaries in Sweden per year. . . . 6
4.1 Cluster example for related residential burglaries. ............. 24
5.1 EULA extractor ............................................. 39
5.2 AUC for different amount of kept attributes ................. 47
6.1 Proposed method concept .................................. 62
6.2 Example of SVM training process. .......................... 70
6.3 AUC comparison of models ................................. 73
8.1 A view of local crimes with red markers denoting similar crimes in the suggested DSS. ............................... 96
8.2 A view of local crimes for a specific search in the suggested DSS. .................................................. 99
8.3 Cluster solution example for spatial proximity. .............. 117
8.4 Correlation plot between Modularity and Rand Index for clustering solutions from Experiment 2. ................. 124
9.1 Spatial and temporal distance between linked and unlinked residential burglary pairs .......................... 138
9.2 Modus operandi distance between linked and unlinked residential burglary pairs .......................... 139
9.3 Calibration plot .................................................. 142
9.4 Law enforcement vs model predictions .......................... 143

10.1 Silhouette index per distance metric for the Spectral clustering algorithm, indicating cluster solution quality. ............... 162
10.2 Rand index per distance metric for the Spectral clustering algorithm, indicating cluster solution accuracy. ............... 162
10.3 Series Rand index per distance metric for the Spectral clustering algorithm, indicating cluster solution accuracy. ............... 163
10.4 Jaccard index per distance metric for the Spectral clustering algorithm, indicating cluster solution stability. ............... 163
List of Tables

4.1 Example comparison of residential burglaries ............... 25
5.1 Feature selection ............................................. 44
5.2 Results for Experiment 2: Random Forest .................... 49
5.3 Results for Experiment 2: Bagging .......................... 50
6.1 Attributes extracted from the Trec07 corpus ............... 68
6.2 Data model comparison ....................................... 72
6.3 Feature selection impact ..................................... 74
7.1 Average clustering measurement and average algorithm rank. 88
7.2 Paired rank comparison of algorithms ......................... 90
8.1 Mean Clustering Measurement for Experiment 1. ............. 113
8.2 Mean Clustering Measurement for Experiment 2. ............. 115
8.3 Fisher’s LSD post-hoc test for Experiment 1 ................. 119
8.4 Fisher’s LSD post-hoc test for edge removal criteria ........ 120
8.5 Fisher’s LSD post-hoc test for combinations of edge representa-

9.1 Distance between characteristics for a randomly chosen linked
pair of residential burglaries. ................................. 140
9.2 Prediction scores of the model on related data. ............... 143
10.1 Summary of labeled crimes and series ......................... 151
10.2 Data characteristics ........................................... 153
10.3 Mean Silhouette index for the algorithms and distance functions 159
10.4 Mean Connectivity index for the algorithms and distance functions ........................................... 159
10.5 Mean Rand index for the algorithms and distance functions 160
10.6 Mean Series Rand index for the algorithms and distance functions ........................................... 160
10.7 Mean Jaccard index for the algorithms and distance functions 160
10.8 Nemenyi test results for Rand index .......................... 164
10.9 Nemenyi test results for Series Rand index ................... 165
10.10 Correlation matrix for the Combined distance metric ........... 166
10.11 Correlation matrix for the Spatial distance metric ............. 166
Law enforcement collects data and physical evidence when investigating a crime scene. There is, however, no systematic and detailed method for collecting the crime scene information. Consequently, the amount of information and which information law enforcement collect differs between crime scenes investigations. Differences can exist not just between different departments but also between crime scenes that the same law enforcement officers examine. The difference is because aspects, e.g. the experience or current conditions of the law enforcement officer, affect the data collection. For instance, a law enforcement officer might be of the opinion that different data is important for two crime scenes.

As a result of the differences in the collected crime scene information, comparisons between cases are very difficult and time consuming to perform. Consequently, identifying serial crimes is difficult, especially with regards to volume crimes. Further, the infrastructure and methods for enabling cooperation between law enforcement counties for detecting serial crimes is lacking. Law enforcement has recently begun to develop methods for sharing and cooperating between counties for volume crimes. There is, however, little support for the cooperation in the IT infrastructure. Further, support in the IT infrastructure for identifying similar crimes committed in other locations (e.g. other cities) or during another time period (e.g. a month earlier) is lacking. Sim-
1. Introduction

Similarly, there is no support for identifying similar crimes committed in other law enforcement counties, as the IT systems are independently deployed. Consequently, any comparison across county boundaries are done via manual comparison involving at least two law enforcement officers investigating crime scene information that might differ vastly.

![Figure 1.1: Entry MO characteristics describe how the perpetrator enters the residence, e.g. by breaking in through a window or a door.](image)

This thesis presents a procedure for systematic collection of crime scene information and a software-based Decision Support System (DSS) for managing the information [2]. Using this procedure, law enforcement collects information concerning e.g. physical position, time, and Modus Operandi (MO) characteristics. An example of two different MO characteristics for entering a building is shown in Figure 1.1. A systematic data collection also enables a more robust comparison of crime scene information in order to detect links between crimes, e.g. residential burglaries. The use of the presented DSS allows easier cooperation during crime scene investigation across law enforcement county boundaries. Further, the DSS makes it possible to search, filter, group, and compare crime scenes with respect to various properties related to modus operandi, location, and so on. This can be seen in Figure 1.2.

The work in this thesis investigate methods to automatically compare and analyze residential burglaries with the aim of aiding law enforcement officers in the process of detecting serial crimes. The goal is
1.1 Aim & Scope

This thesis aims to investigate how descriptive and predictive models can be used for prioritization and filtering of data. The main focus of the thesis is on how descriptive approaches and predictive approaches can be used within law enforcement to analyze and detect links between residential burglaries. By providing means to aid the user in estimating links between cases, law enforcement agents can prioritize resources better and more relevant cases can be investigated. As a consequence,
1. Introduction

Methods for estimating links between residential burglaries can be incorporated into computer-based decision support systems, as filtering, prioritization, or selection tools.

1.2 Outline

In Chapter 2, the background is presented, as well as terminology and related work.

Chapter 3 concerns the approach of the thesis. Section 3.1 lists the research questions. Methodology and Validity threats are presented in Section 3.2 and 3.3.

In Chapter 4, the findings of the thesis is presented and discussed. Section 4.1 concerns the contributions of the publications. The results are discussed and concluded in Section 4.2 and Section 4.3. Section 4.4 presents ideas to future work.

Finally, the publications are presented in Chapter 5-10 and can be considered divided into two parts. First, publication 5 through 7 investigates descriptive and predictive methodology that acts as a foundation for the work in the second part of this thesis. Second, publication 8 through 10 are studies where the lessons learned in Part 1 have been applied to the law enforcement domain according to the aims of this thesis.
Two

Background

Decision support systems (DSS) were first suggested in 1963 as an approach to have computers aid in the decision making process. The term DSS was first introduced in the early 1970s [3]. Multiple surveys on DSS have been conducted spanning research from 1970 – 2001 [3, 4, 5]. Some of the surveys have in common a definition of DSS requiring the following [3]:

- It should support decision-makers rather than replace them;
- It should use data and models;
- It should solve problems with varying degrees of structure, ranging from non-structured to structured problems or problems that are hybrid problems.
- It should focus on the effectiveness rather than the efficiency of decision processes, i.e. sufficient versus optimal performance.

While DSS have been investigated since the 70s, it was not until the mid 1990s that artificial intelligence or machine learning techniques started to become more widely incorporated into DSS [3].

During this time, law enforcement agencies have tried to implement and use different DSS to aid in their work processes. Law enforcement DSS often focus on forensic evidence, which has been divided into two groups, soft and hard forensic evidence. Hard forensic evidence is
2. **Background**

![Graph showing the total number of reported burglaries in Sweden per year.](image)

**Figure 2.1:** Total number of reported burglaries in Sweden per year.

physical evidence, e.g. DNA, fingerprints, and shoeprints. While hard forensic evidence is not always present, soft forensic evidence is always present to a certain degree. Soft forensic evidence is the behavioral aspect of a crime as well as spatial and temporal information, i.e. how was the crime committed, when was it committed, and where was it committed. Whilst there exists standards for collecting hard forensic evidence, one of the problems is that no widely used standards exist for the behavioral aspect of soft forensic evidence. Furthermore, each type of crime would require different types of information collected to be useful. Given these two major constraints, the ability of soft forensic evidence to match crimes have been limited [6].

The amount of crimes committed in what can be called *volume crimes*, e.g. residential burglaries, makes it hard for law enforcement to man-
ually detect patterns and series among the crimes [6]. In 2004, 78% of approximately 5.9 million reported crimes in parts of England and Wales were volume crimes. In Sweden, volume crimes make up 88% of reported crimes according to a report from 2009\(^1\). The amount of reported burglaries has also increased in Sweden over the last years, see Figure 2.1\(^2\). The amount of volume crimes reported makes it a necessity for law enforcement officers to make use of decision support systems to aid in analysis of the crimes. However, the lack of standardized data collection makes pattern analysis difficult.

Geographical information systems (GIS) is one of the more used aspects of decision support systems for law enforcement when investigating volume crimes, and to a lesser extent other crimes. GIS is the mapping of crimes, which might take into account temporal information, allowing easy visualization of spatial crime data. This would allow the detection of geographical hotspots, indicating high-risk areas. Clustering techniques can also be used to detect hotspot or produce groups of crimes that are similar.

A problem when including temporal crime data when analyzing volume crime is that it is often difficult to identify the exact time that the crime occurred [7]. A residential burglary can occur when the owner is away on vacation, and the crime might have occurred any time during the vacation. As such, one must take into account that the crime could have occurred during a time range. Different approaches can be used to approach this problem, e.g. using an aoristic method [7].

Researchers have also investigated using social network analysis to detect and investigate relationships among groups in a criminal network, or investigate connections between friends. Using such techniques, previously unknown connections between persons can be discovered. While classification techniques have been investigated, they

\(^1\)Handling of everyday crimes: A key task for police and prosecutors, http://www.riksrevisionen.se/PageFiles/13727/summary_rir_2010_10.pdf

have produced fairly limited operational results [6]. However, the purpose of applying classification techniques must always be clearly motivated.

2.1 Terminology

Computer Science Terms

*Decision support system* (DSS) helps users make decisions in uncertain situations. DSS can be organized into five groups, extending a categorization from 1980\(^3\) [4, 8, 9]. The five categories are communications-driven, data-driven, document-driven, knowledge-driven, and model-driven. Communications-driven DSS can be exemplified as a system that helps users reach a decision together, e.g., a reputation system. Data-driven DSS can be described as a system that allows easy access to data available in, e.g., files and databases, to help facilitate decision-making. This can be exemplified by real-time monitoring systems or budget analysis systems. Document-driven DSS can be a system that helps users locate correct data, documents, files, or, e.g., web sites. An example of this is a search-engine. Knowledge-driven DSS can be described as a system “that search for hidden patterns in a database”, and can be seen as closely related to data mining [9]. This category requires a good understanding of a specific task. Model-driven DSS uses “data and parameters provided by decision-makers to aid them in analyzing a situation” [9]. Examples of systems include scheduling systems or risk analysis systems. DSS belonging to more than one group are denoted Hybrid DSS.

*Machine learning* concerns the study of programs that learn from experience to improve the performance at solving tasks [10]. Machine

\(^3\)A more extensive look at the earlier categorization and how it relates to the reworked framework can be found online. Included are also additional examples. http://dssresources.com/faq/index.php?action=artikel&id=167
learning comprises a large number of directions, methods, and concepts, which can be organized into learning paradigms. Usually, three paradigms are distinguished; supervised learning, unsupervised learning, and reinforcement learning. The suitability of a certain learning method or paradigm depends largely on the type of data available for the problem at hand.

*Supervised learning* is an area within machine learning that addresses problems based on the existence of predefined classes and labeled data. The labeled data is used to train a model based on patterns in the labeled data. If the data is representative of the population, the model is then able to make predictions on new data [11].

*Unsupervised learning* is an area within machine learning that addresses problems based on unlabeled data to make predictions or classifications. The lack of labeled data makes evaluation of the solution difficult. As such, models are not trained as in supervised clustering. This is often exemplified using clustering, where items are grouped according to e.g. similarity [12].

*Text classification*, or text categorization, concerns the machine learning problem of associating a text document to one or more classes or categories [13]. Text categorization can be used for various purposes e.g. to detect spam [14].

**Law Enforcement Terms**

*Modus Operandi* is a person’s method of operation, i.e. how a person performs a specific action. The term is often used to describe behavioral characteristics in a criminal context, e.g. how victims are chosen [15].

*Volume Crime* are crimes that are committed to such an extent that they impact the community and the local police’s ability of solving the crimes. Often included crime types are street robbery, burglary and
2. Background

vehicle related criminality\textsuperscript{4}.

*Soft forensic evidence* refers to geographical, temporal and modus operandi features of a crime [6].

*Hard forensic evidence* refers to physical evidence, e.g. DNA, fingerprints, etc [6].

2.2 Related Work

Decision support systems (DSS) help users make decisions in uncertain situations. The research conducted has been summarized and reviewed in several surveys since the introduction of the term. These surveys cover the years of 1971-1988, 1988-1994, 1995-2001, as well as a trend analysis through the years 1971-1995 [3, 4, 5, 16]. DSSes require the problem or data to be either structured or fairly structured [17]. The presence of unstructured data, e.g. free-text, requires the decision maker to aid in the process.

DSS have been investigated to solve problems within e.g. the fields of tactical air combat, assisting in stock trading, water resource management, and within the health-care sector, operational assistance, triaging patients and hospital management [18, 19, 20, 21]. For example, research has been conducted on using DSS to help construction tendering processes. Construction tendering processes are an early stage of construction projects dealing with biddings with regard to procurement of services or goods [22]. Even though the use of DSS has been viewed as beneficial to tendering, the current approaches mainly concern structured data and as a consequence do not provide decision support in regards to free-text documents, e.g. contracts [22].

2.2. Related Work

Communications-driven DSS have been implemented for e.g. spam detection and detecting malicious activities in peer-to-peer (p2p) networks [23,24,25]. In the case of spam detection, sender reputation and object reputation were investigated [23,25]. Sender reputation concerned establishing the identity of the sender, which allowed users to rate the identity over time. The problem of sender reputation based spam detection has been identifying the sender, as malicious users forged information or their online presence were short. As a consequence, sender reputation has been useful for honest senders and can be applied to whitelisting approaches [23]. The second approach, object reputation, allowed users to submit fingerprints or signatures of messages considered to be spam. New messages users received were compared against a catalog of message signatures [23]. The problem with object reputation has been the fingerprinting process, as the algorithm should be able to identify variations of messages and at the same time not match legitimate messages [23]. In p2p networks research similar to sender reputation, denoted peer reputation, and object reputation are identified [24,26].

Machine learning based approaches, e.g. clustering or classification can be used to construct knowledge-driven DSS. An example of classification based DSS would be spam detection, where users are unable to process the amount of messages. The first ventures toward automatic spam detection were into automating the rule-based learning techniques [27]. The currently employed anti-spam techniques were summarized [14,28,29]. These studies provide coverage of learning-based spam detection and one of the main conclusions was that automated (machine learning-based) techniques are necessary in order to implement spam filtering.

Knowledge-driven DSS have also been used to problems within the law enforcement domain [30]. With regard to residential burglaries, spatial clustering have been investigated to detect where crimes concentrate in space and time, e.g. to detect hotspots, or to predict future crime locations [30,31,32,33,34,35]. Spatiotemporal correlations over longer time periods have been investigated to further enhance hotspot detection [36].
2. Background

Different hotspot methods are used in DSS for law enforcement agencies, e.g. to detect areas for resource prioritization [34,35]. These approaches differs from crime linkage in that they detect areas which are more likely to have crimes committed, whereas crime linkage finds connections between crimes over larger areas as well [30].

Crime linkage research has focused on crimes conducted that can be considered violent, e.g. sexual offences, rapes, homicides, and different types of burglaries, including violent burglaries [15,37,38,39,40,41]. Different aspects of behaviors can be used for comparison, e.g. MO, spatial proximity, and temporal proximity. Recent research on using MO characteristics have suggested the effectiveness of the characteristics [37]. Research has been conducted into comparison of crimes based on the computed similarity scores, using e.g. logistic regression analysis [38,41,42,43].
Three

Approach

It is difficult to manually detect series of crimes among volume crimes [6]. Automated techniques need to incorporate MO characteristics in order to differentiate between different series. The use of MO characteristics has mostly been limited to link estimation between pairs of crime cases [15,37,38]. Research into clustering crime cases has focused mostly on spatial characteristics, and does not focus on detecting series. The use of MO characteristics put requirements on the quality of the data collected and used [15,41]. Research so far has often limited the data used in the geographical area, and timespan. Further, the data have often been extracted from police databases and coded into a format that is suitable (which can introduce an translation bias). Consequently, only cases where all the behavioral information is available can be used.

The MO characteristics of the data convey different aspects, e.g. method of entry. Link estimation research suggests that MO characteristics does not have equal importance when identifying linked crimes, i.e. method of entry might be more important than which goods have been stolen [15,37,38]. The data should be structured to incorporate different aspects of MO characteristics. When investigating clustering solutions to identify series, the clustering approach needs to take the unequal importance of the characteristics into account.
3. Approach

Due to the low clear up rates for residential burglaries, there are often no solution sets to indicate which crimes are linked. Because of the lack of solution sets, accuracy is not always applicable when evaluating cluster solutions. Other cluster validity measurements need to be investigated for use in a law enforcement domain.

The work in this thesis contributes to the Data Science and Machine Learning domains by investigating methodology for structuring and weighting features, and evaluations of clustering solutions. Further, the thesis presents applied contributions to the fields of Law Enforcement and Crime Linkage regarding methods that can be used to aid the investigative process.

3.1 Research Questions

RQ I. Using supervised machine learning techniques, to what extent can links between residential burglaries be detected?

Law enforcement agencies would benefit in their investigation process if using an automatic system for estimating whether crimes could have been committed by the same perpetrator. Recent research has investigated the use of linear regression analysis to estimate links between cases. Link estimation is investigated in Chapter 9. Methodology applied in Chapter 9 was also investigated in Chapter 5 and Chapter 6.

RQ II. Using unsupervised machine learning techniques, to what extent can residential burglaries be grouped to aid in selection or prioritization of crimes to investigate?

Spatial clustering has, in related research, been investigated to group crimes for investigation. Spatial clustering, however, do not take into account the modus operandi of the perpetrator or the fact that professional criminals can operate over a large geographical area. The ability to filter crimes based on modus operandi would allow law enforcement agencies inves-
3.2 Research Methodology

The research approach applied in this thesis is based on quantitative methods such as quasi-experiments. The specific methodologies and their applications are described in detail in the included articles.

*Experiments* constitute a quantitative research approach “to test the impact of a treatment (or an intervention) on an outcome” [44]. This requires that factors affecting the experiment can be controlled. Experiments can be used to compare for instance the performance of different techniques [44,45]. Experiments use random assignment of study units, e.g. people, to ensure that the study units do not affect the outcome instead of the treatment [46]. *Exploratory data analysis*, or explorative research, is used to investigate little-understood problems, visualize data, and develop questions and hypothesis used in *confirmatory data analysis* methods [47]. Experiments and quasi-experiments are confirmatory data analysis methods focusing on the testing of a hypothesis.

*Quasi-experiments*, compared to experiments, exclude “random assignment of study units to experimental groups”, but are otherwise similar [46,47]. Random assignment is sometimes not optimal due to e.g. constraints concerning cost, participants, or the design of the experiment [46]. Chapter 5 and Chapter 6 use experiments to compare the performance of using machine learning algorithms to differentiate between unsolicited and solicited software. In Chapter 5 the impact of feature selection algorithms is investigated as well. Chapter 8 and Chapter 10 uses controlled experiments to investigate the effectiveness
3. Approach

of clustering to filter/select residential burglaries when investigating series of residential burglaries. Similarly, chapter 9 use experiments for investigating link estimation between residential burglaries.

### 3.3 Validity Threats

Threats to validity can be divided into four main groups: internal, external, construction and conclusion [44,45]. Each group contains several threats, which sometimes, might not be applicable in all research designs [47].

**External** validity threats concern the generalizability of the results. Even if the outcome is true in an experiment setting, the same outcome might be false for a larger scale or in a real world settings [45]. The nature of the data set investigated in Chapter 6 impacts the generalizability, as it contains only emails delivered to a server during one months time and as a consequence it needs to be studied further. This is also a concern for Chapter 8 through 10 as they involve data based on human behavior. Such behavior might always change over time or differ between larger geographical areas, e.g. countries. Chapter 9 uses models trained on this data, and consequently needs to be updated regularly as new data becomes available to reflect new behavioral patterns.

**Internal** validity concerns experimental procedures. Most internal validity threats concern changes in environment and in participants, and that such changes affect the outcome of the experiment [47]. The research investigated in this thesis is of the nature that many of the threats do not apply. Related to this thesis, internal validity threats can be exemplified by the selection threat, meaning that the selection of the population affects the results [44]. This can often be avoided by relying on random sampling from the population. In Chapter 5, Chapter 6, Chapter 8, Chapter 9, and Chapter 10 this is mitigated by using random sampling or cross validation. However, it should be noted that in Chapter 9
and Chapter 10 the data consists of cases that law enforcement agencies have been able to solve and, consequently, the results might be biased towards the population reflected in the data set.

Construction validity threats are the result of inadequate definitions and measurement of variables, e.g. variables defined well enough to be measured [44, 45]. This is less of a problem in any of the included publications as the data measured is not open to interpretations, i.e. labeled data is available. The data is measured using, within the domain, standardized and accepted measurements. However, it should be noted that in Chapter 9 part of the investigation is conducted using unlabeled data. Using unlabeled data is problematic since there is no known answer to evaluate against.

Conclusion validity threats concern inaccurately drawn conclusions from the data [44, 45]. This is also known as statistical conclusion validity [44]. Examples relevant to this thesis are, e.g. low statistical power or violated assumptions of statistical tests. The first is approached in Chapters 5 through 8 by having a large sample size to base our conclusions on. Throughout this thesis, where applicable, standardized statistical tests are used.
Four

Results

4.1 Contributions

The contributions section is grouped into two parts. Part one presents the contribution of the publications from a methodological perspective, investigating methodologies for various purposes. Part two presents the contributions of the publications from a domain-centric perspective, where the lessons learned from part one are applied.

Part One

Chapter 5, titled Informed Software Installation through License Agreement Categorization, presents an automatic prototype for extraction and classification of End User License Agreements (EULAs). Previous research has investigated EULA based classification to detect spyware [1]. Multiple machine learning algorithms have been compared with a state-of-the-art tool [48]. However, the previous research conducted requires user interaction when gathering the EULA, which can be considered infeasible in a large-scale setting. Performance tuning have also been overlooked in this context, which have been beneficial in other cases [49]. The publication investigates methods that can be applied to the problem of RQ I,
4. Results

in that it provides an automatic way of extracting structured data for use in classifying EULA from software. Further, the chapter investigates the impact of feature selection, potentially increasing the performance. The results suggest the applicability of license agreement categorization for realizing informed software installation.

Chapter 6, titled Social Network-based E-mail Classification, presents an approach to detecting unsolicited e-mail messages using several data sources. Previous research have investigated the use of E-mail classification by using previous E-mail conversations to create a correspondence graph, and from that graph, creating a model for classification [50, 51, 52, 53]. Most of the research so far has focused on building social networks from e-mail data, instead of gathering data from OSNs. By the use of other OSN sources as the basis of the classification, it is possible to address the problem of having a large E-mail based history. Thus enabling extended classification for new users, given that said information is available on other OSN. Online social network characteristics are extracted into features that define similarity, or more correctly a level of closeness between users. The features are used to construct a model for spam classification. The constructed model is then compared to traditional spam classification. The results suggested in this chapter answers RQ I and RQ II, investigating the collection of feature selection and model construction for spam classification. Further, it allows users to prioritize messages using the structured data. This could also be potentially adapted and used to prioritize related residential burglars based on communication patterns.

Chapter 7, titled Comparison of Clustering Approaches for Gene Expression Data, evaluates multiple clustering algorithms applied to the problem of clustering genes. Clustering techniques have been one of the methods investigated to identify patterns of gene expressions, with the purpose of allowing an increased understanding of the function of gene expressions or relationships between gene expressions [54, 55]. Different evaluations of algorithms have applied different cluster evaluation metrics on different data sets [54, 55, 56, 57, 58]. The use of different metrics
4.1. Contributions

and data sets make comparing evaluations of algorithms non-trivial. The algorithms are evaluated over multiple related data sets containing time series of genes growth. Multiple cluster validity measurement are investigated to evaluate the produced clustering solutions. The results are evaluated using Friedman’s test and Nemyeni post-hoc test. This helps answer RQ II, in that it investigates a method for evaluating which algorithm is the best for a specific problems and cluster validity metrics to use for such an evaluation.

Part Two

Chapter 8, titled Detecting Serial Residential Burglaries using Clustering, investigates the use of clustering residential burglaries based on MO, spatial, or temporal characteristics. Spatial proximity have been investigated for use in groupings of crimes to detect hotspots [30,31,32,33,34,35]. Recent research on using MO characteristics have suggested the effectiveness of the characteristics to detect connections between crimes [37, 41, 43]. However, using MO characteristics have not been investigated for clustering residential burglaries. The MO characteristics are constructed from crime scene report data. Clustering residential burglaries based on MO would allow analysts to detect and select cases where it is likely that the same perpetrator has been involved over a larger geographical area, and across several counties. The ability to produce cluster solutions based on different MO characteristics is evaluated and compared against spatial and temporal characteristics. The results of this chapter partly answers RQ II, in that it suggests that the feasibility using certain MO characteristics for clustering as an alternative to spatial data. The contribution of the paper is the investigation of the clustering accuracy of MO characteristics, suggesting that the choice of which characteristic to use when grouping crimes can positively affect the end result.

Chapter 9, titled Linking Residential Burglaries, investigates the practical use of logistic regression modeling for use in estimating the probability that two cases are linked. Law enforcement officers often compare
4. Results

case reports against previously reported cases to find common characteristics that might indicate shared perpetrators. The ability to estimate the probability that two cases are linked automatically would allow law enforcement to drastically decrease the time spent on case comparison. The problem of linking reported crimes have been investigated previously. Most research into linking cases have focused on crime types of serial characteristics, often with violent aspects, e.g. sexual offences, rapes, homicides, and different types of burglaries, including violent burglaries [15,37,38,39,40,41]. It is interesting to reproduce earlier research, as it has been conducted on a sample from a small geographical area in, e.g. the UK, and “the utility of other predictors may vary across different geographical areas and different samples” [15,41]. Further, the data in earlier studies are extracted from unstructured crime reports, which might be incomplete or contain biases. This chapter answers RQ I. The results suggest that under favorable conditions, this would allow law enforcement officers to reduce the time spent on comparing cases. The contribution of this paper is the extended investigation into using regression learners to estimate link probability over a large geographical sample. Further, the practicality of logistic regression analysis for estimating link probability of Swedish residential burglaries is investigated.

Chapter 10, titled Combining Modus Operandi for Clustering Burglaries, investigates the use of a distance metric based on the combined MO, spatial, and temporal characteristics. The work can be considered a continuation of Chapter 8. Pair wise link estimation found that there are reasons to weight and combine multiple characteristics [41,43]. This suggests a potential increase in the accuracy of clustering based solutions for grouping residential burglaries. The chapter investigates whether a combination of residential crime characteristics would provide a better accuracy for the clustering solutions. The results of this chapter partly answers RQ II, in that it suggests that certain combined residential burglary characteristics cluster data with similar or better accuracy than spatial data. The chapter also investigates and evaluates the performance of multiple clustering algorithms. The contribution of this paper is the investigation into a distance metric that uses combined crime
characteristics for clustering residential burglaries.

4.2 Discussion

The techniques investigated in this thesis belong to the category of knowledge-driven DSS. The goal was to use knowledge of the problem domain as the basis for the suggested decision, either through grouping of cases based on similarity or through pattern detection in cases based on an understanding of the problem. While a knowledge-driven DSS is capable of giving estimations suggesting a course of action, there is always a possibility of errors. As this thesis concerns primarily DSS aimed at assisting investigation and intelligence work for law enforcement, the suggestions of the DSS can potentially steer the investigation in the wrong direction. Consequently, the final decision must still be made by the decision-maker and can thus only be suggested by the system [9].

The DSS systems presented and investigated should instead be seen as advisory systems capable of decreasing the work burden of law enforcement officers. The amount of volume crimes means that significant crime patterns is likely to remain undetected with a manual investigation, making DSS systems a necessity [6]. As the workload of law enforcement officers is decreased, an increase in the cases investigated, their relevance, and possibly solved is expected. By using a clustering approach based on not only spatial and temporal data, but also MO information, as an initial selection tool law enforcement officers would be able to more efficiently decide which cases to focus on when investigating a series, as is exemplified in Figure 4.1. This could be further aided by using a classification DSS to estimate the probability that the cases in the cluster is committed by the same perpetrator(s). This would allow efficient comparison between cases not necessarily close in time and space.

Chapter 9 investigated link estimation between cases and evaluated,
Figure 4.1: Cluster example for residential burglaries. Similar burglaries are connected and known series shown.

among other things, the time required for law enforcement officers to manually estimate linkage compared to a DSS. While the DSS should only be seen as an indicator or advisory system, the results still suggested that the time spent on estimating links could be greatly reduced. This would allow law enforcement officers to focus more resources on related crimes or to investigate other cases. It should be noted that while the use of such a system could still fail to detect links between cases, law enforcement officers employ the manual equivalent very sparingly. Due to the amount of cases available, links are often not detected [6].

Law enforcement officers often only compare cases that are spatial and temporal close, or where there is some other indication that the cases are linked. Spatial-temporal characteristics are often not enough to detect links [6]. A problem with a spatial/temporal approach is that
4.2. Discussion

Table 4.1: Example comparison of characteristics for a randomly chosen, linked pair of residential burglaries.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal</td>
<td>79 days</td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td>11.383 kilometers</td>
<td></td>
</tr>
<tr>
<td>Combined</td>
<td>0.380</td>
<td></td>
</tr>
<tr>
<td>Entry</td>
<td>0.483</td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>0.424</td>
<td></td>
</tr>
<tr>
<td>Goods</td>
<td>0.471</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td>0.182</td>
<td></td>
</tr>
<tr>
<td>Victim</td>
<td>0.353</td>
<td></td>
</tr>
</tbody>
</table>

The system can miss links between cases that are committed by criminals over a large geographical area, e.g. multiple counties, or over a long timespan. An example of a comparison can be seen in Table 4.1. The table contains the different MO characteristics distances, spatial and temporal distances, as well as a visual representation of the spatial distance. The MO characteristics are measured using Jaccard distance.

The quality of any linkage between cases depends on the quality of the collected data, accuracy of collection and the features that is collected [6]. The systematic data collection method is an attempt to address this. The features collected are iterated by law enforcement experts regularly, ensuring the relevancy of the features. Most systems today extract the features from existing crime reports which is dependent on the quality of the collection, i.e. not all features might be collected [15,41,59]. The proposed data collection method is used by law enforcement agents at the crime site to collect the specific information, i.e. thus improving data collection accuracy.

Further, the use of the DSS allows law enforcement officers to move away from confirmatory investigations, i.e. an experienced law enforcement officer have an idea of which cases are linked and tries to confirm that idea. Consequently, the use of the DSS would allow an increase in the objectivity of the analysis and investigation process.
Another practical benefit of using the DSS is the ease of finding and initiating information exchange over several police counties. Making comparisons over county boundaries is often limited within Swedish law enforcement due to organizational constraints. Organizational constraints are not limited to Swedish law enforcement [59]. The ability to easily and efficiently share information is often lacking [60]. Investigations across several counties are aided by the implemented social and subscription features, allowing easy sharing of relevant case series and alerts for new cases matching certain criteria, e.g. based on link probability.

One aspect to keep in mind when investigating any crime linkage is that the data used to train and test any approach is based on solved cases. This can be problematic as the data might not be representative for all types of criminals, or contain other biases [15]. It could be that law enforcement officers have a higher percentage of offences solved for certain types of criminals, such as local perpetrators. A consequence of this would be that any classifier based on the solved crimes are biased towards this group of criminals and connection between crimes committed by other types of criminals might be overlooked or given a low probability.

The fact that the model could be biased towards certain types of criminals could also be used to the advantage of law enforcement agencies. Given that the different categories of criminals are made available for each suspect, models could be trained for these different categories. This could give law enforcement officers further indications where to direct their investigation, as local perpetrators require different actions than national or international perpetrators. Currently, the types of criminals are not available, but this approach should be seen as an interesting future work.

As the approaches investigated dealt with instances where classification has a certain degree of uncertainty, e.g. perpetrator information is unavailable or estimations produced are not binary, i.e. the classi-
fications are suggestions [9]. This is similar to the automated system presented in Chapter 5, which was only able to extract EULAs corresponding to approximately 39% of the applications investigated. Such an application should not be used stand-alone, but rather in combination with other techniques. A comparable situation exists for the DSS aimed at law enforcement agencies. In situations where uncertainty is present, to base the suggestion on multiple techniques would be beneficial to the user of the system, in this case law enforcement officers. This resembles the principle behind ensemble learners, i.e. weighting and combining several opinions to provide increased accuracy [61].

Given that the DSS can estimate erroneous links between cases, the wrong person might be indicated as the suspect for a crime. A potential consequence of such an implication might be that the privacy of a person might be compromised. However, this is an extreme case that could also occur during an investigation that does not make use of the proposed DSS. Further, law enforcement officers should find corroborating evidence to validate the suggested links [9].

The information available in the DSS, whilst indicating the spatial location of a crime scene, is not privacy invasive. First, the access to the information is limited to specific law enforcement officers. Second, the spatial information is the only information that could potentially be linked to a person. Third, Swedish law enforcement has procedures for removing information from databases regularly. The rules for how information can be registered and managed is regulated by Swedish law\textsuperscript{1}. There are, consequently, regulations for access to information, management, and removal of information from law enforcement systems.

\textsuperscript{1}The laws in question is \textit{polisdatalagen} (1998:622) and \textit{personuppgiftslagen} (1998:204). Personuppgiftslagen is a subsidiary law to polisdatalagen.
4. Results

4.3 Conclusion

The work in this thesis investigate the use of DSS to manage and automate police crime investigations, primarily with regards to the investigation of series of residential burglaries by law enforcement officers.

The work presented, even if not directly applied to law enforcement, exemplifies different approaches that machine learning can be used to support decision making processes. By collecting data using a systematic approach, automated comparisons and evaluations can be performed. A method for systematic collection of crime scene information is presented. Chapter 8 through Chapter 10 suggests the possibility of using machine learning based methods to aid law enforcement officers analysis of residential burglaries with regards to detecting series. Law enforcement officers can greatly reduce the time put into analyzing a residential burglary. Further, by using a combination of the techniques suggested in this thesis cases can be filtered and the estimated probability cases are linked can be presented to law enforcement officers. The filtering stage could provide previously unknown related cases that a manual investigation would not have discovered. And, by estimating the probability that cases are linked, law enforcement officers can easily prioritize the cases they investigate.

Automated approaches allows for a more objective investigative process, while also saving resources. As investigators can easily search and filter cases across police counties, information exchange is simplified. Consequently, law enforcement officers are given tools that enable a more robust and uniform investigative process across police counties.

4.4 Future Work

The creation of regression models for different types of residential burglars would be an interesting future work. This would further aid law
4.4. Future Work

enforcement officers work by helping further narrow the amount of potential suspects, as well as provide further information of where to focus the investigations. The investigation of how to systematically collect anonymized data concerning convictions from courts for use in the DSS should be investigated. Similar, whether online social networks could be used for suggesting potential co-burglars is an interesting approach that merits further investigation.

Further, investigating series in other types of crimes, e.g. muggings, has potential to solve volume crimes with low clear-up rate. The ability to detect series of crimes across different crime types would also aid law enforcement officers.
Informed Software Installation through License Agreement Categorization

Anton Borg, Martin Boldt and Niklas Lavesson

Abstract
Spyware detection can be achieved by using machine learning techniques that identify patterns in the End User License Agreements (EULAs) presented by application installers. However, solutions have required manual input from the user with varying degrees of accuracy. We have implemented an automatic prototype for extraction and classification and used it to generate a large data set of EULAs. This data set is used to compare four different machine learning algorithms when classifying EULAs. Furthermore, the effect of feature selection is investigated and for the top two algorithms, we investigate optimizing the performance using parameter tuning. Our conclusion is that feature selection and performance tuning are of limited use in this context, providing limited performance gains. However, both the Bagging and the Random Forest algorithms show promising results, with Bagging reaching an AUC measure of 0.997 and a False Negative Rate of 0.062. This shows the applicability of License Agreement Categorization for realizing informed software installation.
5. Informed Software Installation through License Agreement Categorization

5.1 Introduction

This work addresses the problem of uninformed installation of spyware and focuses on analysing End User License Agreements (EULAs). Malicious software (malware) vendors often include (disguised) information about the malicious behavior in the EULAs to avoid legal consequences. It would therefore be beneficial for the user to get decision support when installing applications. A decision support tool that can give an indication whether an application can be considered spyware or not would presumably make the installation task simpler for regular users and would enable the user to be more secure when installing downloaded applications. We present an automated method that extracts and classifies EULAs and investigate the performance of this method. More concretely, the proposed method is based on the use of machine learning techniques to categorize previously unknown EULAs, as belonging to either the class of legitimate or malicious software. Machine learning, in this context, enables computer programs to learn relationships between patterns in input data (EULAs) and the class of output data (malicious or legitimate software). These relationships can be used to make classifications of new (unseen) EULAs.

5.1.1 Aim and Scope

The primary aim of this study is to present a method for automatic EULA extraction and classification. Additionally, we, using this method, obtain and prepare a large data set of EULAs. This data set is used for benchmarking four different algorithms. Evaluating the impact of feature selection and machine learning algorithm parameter tuning is also done\(^1\).

\(^1\)A web link to the actual database will be provided in a potential camera ready version

32
5.2. Background and Related Work

5.2.1 Background

Malware, e.g. viruses, originated from a rather small set of software, with the primary goal of generating revenues for the attacker, or creating chaos among infected computer systems [62]. To protect users from these types of software, anti-virus tools emerged. As the malware at this point were illegal, removing malware was a question of using the resources and techniques available at the time [63].

At the end of 1990, a new type of malware emerged, known as spyware, with the purpose of gathering personal information. Due to the increase of the number of Internet users, a market for targeted online advertisement developed. Spyware was not considered explicitly illegal, which complicated malware removal and resulted in the creation of a legal grey zone.

A common technique for detecting malware was to blacklist these applications through the use of signatures, i.e., by statically dividing between legitimate and malicious software. However, this required a copy of the malware to first be captured on the Internet in order to
5. Informed Software Installation through License Agreement Categorization

create a unique signature, and then being distributed to all customers of the anti-virus tool [63]. The main drawback with this technique is the fact that the anti-virus tools were one step behind the creators of malware. Another drawback is related to the vast amount of malware that spread on the Internet, increasing the size of the signature database rapidly and resulting in significantly decreased performance when used by customers.

Anti-virus manufacturers therefore began researching alternative techniques for solving the problem. For example, agent-based approaches [64] and artificial neural networks [65] [66] was investigated. Another technique used was dynamic analysis, which kept a suspicious program in captivity within a so-called sandbox, e.g. a virtual machine, while monitoring its execution as a way to discover any deviant behavior [67] [63]. Even though dynamic analysis could be used for computer viruses, e.g. by detecting the self-replication routines, it was much harder to distinguish spyware or adware applications from their legitimate counterparts. The reason is that adware and spyware applications simply show information on the screen or transmit rather small quantities of data over the network, i.e. behaviors that are shared by most legitimate programs as well.

5.2.2 Machine Learning

The machine learning discipline concerns the study of programs that learn from experience to improve the performance at solving tasks [10]. A large number of directions, methods, and concepts, which can be organized into learning paradigms. Usually, three paradigms can be distinguished; supervised learning, unsupervised learning, and reinforcement learning. The suitability of a certain learning method or paradigm depends largely on the type of available data for the problem at hand.

From a machine learning perspective, the main problem studied in this paper can be described as that of learning how to classify software
applications on the basis of their associated EULAs by generalizing from known associations of EULAs and software application classifications [1].

More formally, and based on suitable definitions [68], we assume the existence of a universal set, $I$, of EULA instances. Each EULA instance, $i \in I$, is defined by a set of features (e.g., words, strings, values, and so on). Furthermore, we assume that the EULAs can be categorized into a limited number of categories or classes, $C$.

The learning task is then to generate a function (or mapping) between $I$ and $C$. This function, $c : I \rightarrow C$, is known as a classifier or generalization. In practice, however, one does not have access to the complete set, $I$, or the correct classification of each element of that set. Instead, a common case is to have a limited set, $J \subset I$, of instances (inputs) and correct classifications (outputs).

Thus, the practical objective, is to generate a classifier by generalizing from $J$ (or a subset of $J$) and the associated known classifications for each instance of $J$. Since we are interested in generating a classifier that will indeed be able to classify unknown instances (instances from $I$ but which are not included in $J$), we need to estimate the theoretical classification performance on $I$ by calculating the classification performance of $J$ (or, again, a subset of $J$).

The common practice in data mining and machine learning is to divide $J$ into two distinct sets; the training set, $J_{\text{train}}$, and the testing set, $J_{\text{test}}$. This way, a classifier can be generated from $J_{\text{train}}$ and the prediction performance can be estimated by computing the classification performance on $J_{\text{test}}$.

There are many learning algorithms available that can perform the task of generating a classifier from input data associated with known outputs. However, each algorithm has its own learning bias, which is used to define the search space (the set of available classifiers) and the traversal of the search space. A completely unbiased learner would have
access to the complete set of possible classifiers and would be able to
traverse this set in any possible way. Of course, this search is practically
infeasible in real-world situations. It is therefore necessary to select an
algorithm, or a set of algorithms, whose learning biases are most suitable
for the problem at hand.

5.2.3 Related Work

Previous research on the use of text classification techniques within the
context of EULAs is quite sparse. It have been shown that it is possible
to use machine learning techniques to address the problem of EULA
classification [1] [48]. State-of-the-art within commercial tools involve
one stand-alone application called EULAlyzer and one website called
EULA Analyzer that includes the ability to analyze a EULA. Both of
these services are proprietary and therefore lack information regarding
their design and internal construct. However, it seems as if they make
use of blacklisted words to simply highlight any sentence within a EULA
that contains one of the blacklisted words. The computer user then has
to read through the highlighted sentences to try to come to a conclu-
sion whether the particular EULA should be considered legitimate or
spyware.

A comparison is made between EULA Analyzer and 15 machine
learning algorithms [48], with the conclusion was that both the Support
Vector Machines and Naive Bayes Multinominal algorithms performed significantly better than the state-of-the-art tool. Finally, it
could also be added that the performance of these two algorithms have
later been improved even more when utilized on an extended data set of
EULAs [1]. However, the previous research conducted requires user in-
teraction when gathering the EULA, which can be considered infeasible
in a large-scale setting. Performance tuning have also been overlooked

---

in this context, and should be investigated as it has proven beneficial in other cases [49].

5.3 Approach

We have gathered 7,041 applications, where approximately 21% of the applications are malicious, from which we extract EULAs to form an extended data set. The EULAs were extracted using the automated tool described in Section 5.3.1. The malicious applications, counting 1,530 applications, have been provided by Lavasoft\(^4\) and the legitimate applications, counting 5,511 applications, have been downloaded from CNET’s download.com site\(^5\). Download.com thoroughly checks for malware among the applications made available to the public and can thus be said to be a good source of legitimate applications.\(^6\) We also, in Section 5.4, test the data of the two sources to find differences.

In order to extract the licenses from the applications, we make use of the automated system presented in section 5.3.1. From the applications we managed to extract a number of license agreements for use in our experiments. As not all applications have licenses and our extractor does not support all file structure, as described in Section 5.3.1, this has left us with 810 malicious licenses and 1,961 legitimate licenses. For software with localized EULAs, only the English version were kept. These numbers means that our automatic tool is currently capable of extracting licenses from approximately 53% of the malicious applications and 35% of the benign applications. With more extractors implemented, this number is likely to increase. Many benign applications have similar licenses, but since minor details still vary and they help with the categorization, similar licenses are kept.

\(^4\)Ad-Aware by Lavasoft, http://www.lavasoft.se
\(^5\)CNET Download.com, http://download.com
5. Informed Software Installation through License Agreement Categorization

5.3.1 Automated System

We have for this study developed a system for automated license classification, using a binary file as in-parameter and presenting the user with a classification of the binary file based on the EULA. Earlier research has used a manual process of extracting EULAs from the binary, which is infeasible in a real world setting. We have implemented an automated system using machine learning techniques for this purpose. In its current incarnation, it supports the standard installer types, e.g. NSIS, MSI, Inno setup, as well as standard archive formats, and makes use of publicly available programs as subsystems. Our proposed system is divided into three stages, extraction, transformation and classification. A flowchart of how the proposed system works can be seen in Figure 5.1 and pseudo code for the identification stage is shown in Algorithm 1. The system is implemented in Ruby\(^7\) and in a way designed to make it easy to extend, thus adding support for more types of applications is fairly simple.

The system is based on the premise that a wide range of installers are roughly equivalent to a compressed archive. In order to know which extraction routine to use, the system identifies the binary file. To do this, the system makes use of the program TrId\(^8\). The system then tries to extract the EULA from the binary file. Depending on the result of the identification, this is done using different extraction routines. An example of this is the MSI installer. MSI installers store licenses in Rich Text Format (RTF) inside their string data. The system locates the RTF data inside the string data file and extracts it. To perform the decompression of the binary file, we have built our system around the 7zip extractor\(^9\). 7zip supports a large number of filetypes and are thus suitable for our system.

The transformation stage is divided into two substages, conversion

---

\(^8\)Mark Pontello’s Home, http://mark0.net/soft-trid-e.html
\(^9\)7-Zip, http://www.7-zip.org/
Figure 5.1: A conceptual view of EULA extraction and classification.
5. **Informed Software Installation through License Agreement Categorization**

---

**Algorithm 1** File Identification

```
1: function check(path)
2:     if path is a file then
3:         type = typeIdentifier(path)
4:         if type is an installer then
5:             Extractor(path, type) *
6:         else if type is a document then
7:             if path.name contains license or eula then
8:                 SaveDocument(path)
9:         end if
10:     else
11:         extracted = Extractor(path, default)
12:         if extracted is not NULL then
13:             for all file in extracted do
14:                 check(file)
15:         end for
16:     end if
17:     end if
18: else if path is a directory then
19:     for all file in path do
20:         check(file)
21: end for
22: end if
23: end function
```

* The extractor chooses a suitable extraction routine based on the type. The extraction of an MSI installer is described in Section 5.3.1.
and preprocessing. In the conversion stage, the system converts the license to plain text. This is for the preprocessing to be able to read the license agreement, as the different installers store the license agreements in different formats, e.g. MSI uses the RTF format. The RTF licenses for example is converted to plain text files using the UnRTF\textsuperscript{10} program, stripping everything but the text from these files. UnRTF is specifically designed to convert from RTF to other formats. The preprocessing sub-stage is described thoroughly in section 5.3.2. The result of this stage is a EULA instance that the classifier can categorize.

The EULA instance is then passed to the classifier stage where it is categorized using machine learning algorithms. The result of the categorization is then presented to the user, helping the user to decide if the application is either good or bad, and whether or not to install the application.

### 5.3.2 Data Preprocessing

We conduct our experiment using the Weka machine learning workbench [71], a commonly applied suite of algorithms and evaluation methods. In order for Weka to be able to process the EULAs, we have opted to remove special characters and to only keep the standard latin characters. As few machine learning algorithms can process strings, we transform the strings to a more suitable representation. Ways for representing text include, e.g.: meta data (such as word length, frequency or the number of words) [72], bag-of-words (where each word in the text is defined as a feature) [73] and \(n\)-grams [74]. [73] also looks at phrase based features, where words would form a phrase which is able to better convey the meaning of the sentence, and Hypernym based features, where relationships between words is taken into account. The study found that the bag-of-words model outperformed the more complex text representation methods [73]. In the bag-of-words model, strings are tokenized to

\textsuperscript{10}UnRTF, http://www.gnu.org/software/unrtf/unrtf.html
words and represented by word vectors. In Weka, this transformation is carried out using the StringToWordVector filter, which we apply to the licenses. We employ the following filter configuration: a maximum of 2,000 words are stored per category, TF IDF (Term Frequency-Inverse Document Frequency) is used for word frequency calculation, and the Iterated Lovins stemmer is used to reduce the number of words by keeping only the stems of words.

Software licenses can contain a large amount of text and as a result, yields a large number of features. Many of these features are not useful to the learning algorithm and have, in some cases, even been shown to deteriorate the performance of the classifier [71]. We have therefore chosen to remove some of the attributes left by the StringToWordVector filter.

Feature Selection is the process of reducing the number of features available in a data set in order to increase either the classification and/or the computational performance [75]. This is done by, using an algorithm, removing features that are deemed unlikely to help in the classification process. It has been shown that classification accuracy have been improved when reducing the number of features using feature selection algorithms [76]. Several comparisons between feature selection algorithms applied on text categorization have been done in the past. The results is that $\chi^2$ is often considered to be the most efficient algorithm [75] [76]. However, when compared to Categorical Proportional Difference(CPD), CPD have been shown to outperform traditional feature selection methods, e.g. $\chi^2$ [49]. As a result we choose to use CPD as the feature selection algorithm of our choice. However, as we do not know which is the best cutoff point, i.e. how many attributes CPD should remove, we have defined a keep ratio interval and selected a step size. In the presented study, we use a keep ratio interval of 100% to 10% together with a step size of 10%. After applying CPD, we are left with 10 data sets, where the attributes range from 10% of the attributes kept to 100% of the attributes kept.
5.4 Experimental procedure

We want to determine whether the classification results obtained in previous research are valid for a larger data set. We also investigate if the classification performance can be increased using feature selection or by tuning problem-specific algorithm parameters.

Four algorithms have been chosen as a basis for our experiments. The algorithms are Bagging, Random Forest, Naive Bayes Multinomial and Support Vector Machine (SVM). The three first were selected on the basis of previous experimental results [1], and we chose to include only one algorithm from each family of algorithms. SVM was also included since it has been proved to work well in other text categorization tasks [77]. In both experiments, the performance is estimated by using the 10-fold cross-validation test. 10-fold cross validation is the process of dividing the data set into 10 subsets (folds), using 9 folds for training and 1 fold for testing. This is then repeated 10 times, switching the testing fold each time. Before running our experiments we executed preliminary experiments, which indicated that feature selection combined with parameter tuning do not yield any specific performance boost when used together. Therefore, we conduct separate experiments for feature selection and parameter tuning. Also, we made an attempt to validate that the learning algorithms included in our experiments indeed detect the differences between benign and spyware EULAs. Therefore, we divided the data set into two dummy classes that each included 50 % of the benign EULAs and 50 % of the spyware EULAs. Then we used the Naive Bayes Multinomial learning algorithm to generalize from these two dummy classes to make sure there were no patterns separating them, i.e. patterns included in both the randomly selected benign and spyware EULAs. This resulted in a AUC score of 0.526, which is very close to random guessing (AUC is explained in Section 5.4.3). Therefore the results indicate that there does not seem to be any other hidden patterns tying the classes together, and thus that our data set is valid for further exploration.
5. **Informed Software Installation through License Agreement Categorization**

<table>
<thead>
<tr>
<th>Feature Set Size$^a$</th>
<th>Weighted AUC</th>
<th>SVM</th>
<th>Bagging</th>
<th>Random Forest</th>
<th>NB$^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Absolute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10% 140</td>
<td>0.929</td>
<td>0.932</td>
<td>0.941</td>
<td>0.802</td>
<td></td>
</tr>
<tr>
<td>20% 278</td>
<td>0.946</td>
<td>0.976</td>
<td>0.986</td>
<td>0.863</td>
<td></td>
</tr>
<tr>
<td>30% 417</td>
<td>0.964</td>
<td>0.989</td>
<td>0.995</td>
<td>0.891</td>
<td></td>
</tr>
<tr>
<td>40% 555</td>
<td>0.968</td>
<td>0.991</td>
<td>0.994</td>
<td>0.956</td>
<td></td>
</tr>
<tr>
<td>50% 693</td>
<td>0.968</td>
<td>0.992</td>
<td>0.996</td>
<td>0.973</td>
<td></td>
</tr>
<tr>
<td>60% 832</td>
<td>0.975</td>
<td>0.990</td>
<td>0.993</td>
<td>0.956</td>
<td></td>
</tr>
<tr>
<td>70% 970</td>
<td>0.978</td>
<td>0.992</td>
<td>0.997</td>
<td>0.936</td>
<td></td>
</tr>
<tr>
<td>80% 1,109</td>
<td>0.977</td>
<td>0.991</td>
<td>0.994</td>
<td>0.903</td>
<td></td>
</tr>
<tr>
<td>90% 1,247</td>
<td>0.977</td>
<td>0.995</td>
<td>0.995</td>
<td>0.896</td>
<td></td>
</tr>
<tr>
<td>100% 1,385</td>
<td>0.977</td>
<td>0.995</td>
<td>0.992</td>
<td>0.897</td>
<td></td>
</tr>
</tbody>
</table>

$^a$ The fraction of attributes to keep after CPD attribute ranking

$^b$ Naive Bayes Multinominal

Table 5.1: Feature Selection

5.4.1 **Experiment 1: Feature Selection**

In this experiment we investigate what effects feature selection have on the performance results of the four machine learning algorithms mentioned previously. In order to evaluate any potential performance gains we create ten new data sets that all are subsets of the initial data set containing 2,771 EULAs. This gives us data sets ranging from 10 to 100% of attributes kept, with a step size of 10%. The number of attributes for 10% and 100% are 140 respectively 1,385, as can be seen in Table 5.1.

5.4.2 **Experiment 2: Parameter Tuning**

In the second experiment we investigate if it is possible to increase the performance using parameter tuning of the two algorithms with the highest performance measure from the previous experiment. The two algorithms included in this experiment were Bagging and Random Forests. We opted to select the top two performers rather than the top performer, since these two algorithms both showed fairly similar results. Moreover, the ways in which the two algorithms can be configured are
quite different from each other. The variables that we use for parameter tuning is discussed below. However, it should be mentioned that both algorithms are ensemble algorithms, meaning that they each use several different learning algorithms that votes on the classification of each instance. It is then the task of the ensemble algorithms to reach a decision based on the results from the different learning algorithms used.

**Random Forests**

Random forest contains two main variables that we tune in order to determine if we are able to increase the performance. The first variable is the number of trees created in the forest. Each tree gets to vote for the instance, and the class with most votes is picked. Thus, the higher number of trees, the more votes are used as a basis for the classification. The second variable is the number of attributes used to build each tree. The number of attributes is a subset of all available attributes within the data set, and is chosen randomly for each tree. A higher number of attributes decreases the errors produced by the forest. However, it also makes each tree more similar [78]. For both these values we have chosen a symmetric range of values to use, based on the default values available in Weka.

The default value for the number of trees in Weka is 10. Based on this value, the range we have chosen is between 4 and 16 trees (inclusive) with a step interval of 2. The default value for the number of attributes is $\log_2(n) + 1$, where $n$ is the number of attributes available in the data set. Working from this we have calculated our ranges of values for the number of attributes with $\log_2(n) \pm x$, where $x$ is a range from -3 to 3 with a step size of 1. The number of attributes for the data set with 100% of the features left, seen in Table 5.1, is 1385 and based on this we get that the default value for our data set is 11.
5. Informed Software Installation through License Agreement Categorization

Bagging

In Bagging we have chosen to investigate how tuning the bag size based on the training set used, as well as the number of iterations that the bagging algorithm performs affect the performance of the algorithm.

The first variable is the size of the bags from which the trees in the model is built. The sizes of these bags are set as a percentage of the training set size. The bags are filled randomly by sampling with the replacement from the original training set. This means that in each bag, there will be duplicates, as each instance in the training set can be selected more than once. The second variable is the number of iterations, which decides how many trees should be created within the current bag [79]. The vote result of each tree is then used as the result for the current bag.

The default value for the number of iterations is 10, and we have chosen to investigate the range 6 to 14 with a step interval of 2. The training sizes investigated is an five percent step ranging from 100%, which is the default value in Weka, to 75%.

5.4.3 Evaluation Metrics

We represent EULAs associated with spyware programs as positives, while benign EULAs are represented as negatives in our experiments. Used metrics are True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN). A TP is a spyware instance classified as spyware and a FP is a benign instance classified as spyware. A TN is a benign instance classified as benign and a FN is a spyware instance classified as benign. The True Positive Rate (TPR) and False Negative Rate (FNR) are used to see how the spyware EULAs were classified. TPR is defined as $TP / (TP + FN)$ and FNR is defined as $FN / (TP + FN)$.

When evaluating the performance of the algorithms, we have chosen
5.5. Results

Figure 5.2: AUC for Random Forest, Bagging, Naive-bayes Multinomial and SVM on data sets with different amount of kept attributes.

to use the weighted area under the ROC curve (AUC) single point measure, which is based on TPR and the FPR. Two important properties of the AUC metric is that it is not depend on equal class distribution or misclassification costs [71]. The calculation of, and motivation for, AUC is described in detail in [80].
5. **Informed Software Installation through License Agreement Categorization**

### 5.5 Results

#### 5.5.1 Experiment 1

As can be seen in Figure 5.2, Random Forests and Bagging outperform Naive Bayes Multinomial. The latter performs best when only 50% of the attributes are kept. This suggests the use of feature selection when employing Naive Bayes Multinomial. However, Bagging and Random Forests still perform better than Naive Bayes Multinomial. SVM also performs well, but there is a clear loss of performance as the number of attributes is decreased, and the algorithm does not perform as well as Bagging and Random Forests, as can be seen in Table 5.1.

Bagging and Random Forests perform fairly similarly as long as the percentage of kept attributes is 30% or higher. This shows that it is possible to remove a fairly large amount of the attributes before starting performance is degraded. However, since the feature selection does not provide any performance enhancement, in fact there seem to be a small performance loss in most cases, there is no obvious argument for applying feature selection during pre-processing.

#### 5.5.2 Experiment 2

The second experiment concerned the impact of parameter tuning on Random Forests and Bagging. In the following two subsections, we present the results of our experiments.

**Random Forest**

Table 5.2a and 5.2b shows the result of the parameter tuning done on the algorithms. Looking at Table 5.2a we can see that the performance of the
algorithm correlates to the number of trees used in the forest. However, although an increased number of trees, the actual performance gain, when measured in AUC, is minimal. In Table 5.2b, showing the effects of using different amounts of attributes used when creating each tree, we see that for each number of attributes the results vary. These results do not seem to be related in any way, indicating that one cannot argue that tuning this variable is beneficiary to the overall performance.
5. Informed Software Installation through License Agreement Categorization

<table>
<thead>
<tr>
<th>Bag size</th>
<th>Weighted AUC mean (STD)</th>
<th>FNR mean (STD)</th>
<th>TPR mean (STD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% *</td>
<td>0.995(0.005)</td>
<td>0.061(0.017)</td>
<td>0.942(0.023)</td>
</tr>
<tr>
<td>95%</td>
<td>0.997(0.003)</td>
<td>0.062(0.018)</td>
<td>0.942(0.023)</td>
</tr>
<tr>
<td>90%</td>
<td>0.996(0.004)</td>
<td>0.061(0.022)</td>
<td>0.938(0.022)</td>
</tr>
<tr>
<td>85%</td>
<td>0.996(0.004)</td>
<td>0.068(0.024)</td>
<td>0.934(0.031)</td>
</tr>
<tr>
<td>80%</td>
<td>0.994(0.005)</td>
<td>0.062(0.023)</td>
<td>0.935(0.025)</td>
</tr>
<tr>
<td>75%</td>
<td>0.995(0.005)</td>
<td>0.074(0.010)</td>
<td>0.928(0.021)</td>
</tr>
</tbody>
</table>

(a) Results of tuning the bag size in Bagging algorithm

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Weighted AUC mean (STD)</th>
<th>FNR mean (STD)</th>
<th>TPR mean (STD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.994(0.005)</td>
<td>0.063(0.015)</td>
<td>0.944(0.019)</td>
</tr>
<tr>
<td>8</td>
<td>0.994(0.005)</td>
<td>0.062(0.018)</td>
<td>0.943(0.019)</td>
</tr>
<tr>
<td>10*</td>
<td>0.995(0.005)</td>
<td>0.061(0.017)</td>
<td>0.942(0.023)</td>
</tr>
<tr>
<td>12</td>
<td>0.995(0.004)</td>
<td>0.057(0.020)</td>
<td>0.944(0.024)</td>
</tr>
<tr>
<td>14</td>
<td>0.996(0.004)</td>
<td>0.057(0.020)</td>
<td>0.944(0.024)</td>
</tr>
</tbody>
</table>

Default value is marked with an asterisk. All other parameters are left as per default.

(b) Results of tuning the number of iterations in Bagging algorithm

Table 5.3: Results for Experiment 2: Bagging

Bagging

The effects of tuning the Bagging algorithm is presented in Table 5.3a and 5.3b. The results in Table 5.3a indicates that it is possible to lower the bag size, up till a certain point, and gain performance. However, the results are contradictory and should not be taken as a certainty. However, compared to Random Forest, the TPR and FNR values are worse.

Table 5.3b shows us that increasing the number of iterations do in fact produce a better result, going from 0.995 to 0.996, concerning AUC. The TPR and FNR values indicate that an increased number of iterations provide a minimal performance gain. The TPR in Table 5.3a indicates that decreasing the bag size can result in decreased performance.
5.6. Discussion

We have in this work presented an automated tool that (based on the EULA) decides if an application is considered malicious or benign. The use of this tool would help the user to decide whether or not an application can be considered malicious. In order to make this tool work we had to implement a number of extractors capable of extract the EULA for the program. As there exists several different installer formats and packers, we have in this study focused on the ones prevalent in our data set. However, we have built our tool in such a way that it is quite easy to extend it with more formats. It is also written in a way that makes it possible to use it for bulk tasks (e.g. in a bash script) or calling it from a program (e.g. an antivirus tool), the program is flexible and possible to use in both a server environment and desktop environment. Earlier research has shown the feasibility of using machine learning techniques to perform text classification on EULAs to distinguish between malicious and benign applications. However, earlier tools have required manual input when extracting EULA and submitting it for classification. One of the main contributions of this paper is the presentation of an automated tool that is able to extract EULAs, classify them, and then give decision support. The level of performance reached by the algorithms clearly shows the potential of an automated system for EULA classification.

5.6.1 Data Set Content

This work involves a significantly extended data set of 2,771 EULAs compared to the two previous studies that contained 100 and 996 instances, respectively. By using this data set of increased size it is our intention to more accurately mimic a real world setting. During the work of extracting the EULA texts from both the collected spyware and legitimate programs we could see a clear trend that malicious applications to a higher extent contained EULAs compared to benign applications. In this case 53% of the malicious programs contained EULAs compared to
5. Informed Software Installation through License Agreement Categorization

36% of the benign programs, which strengthens the claim that developers of malicious programs make use of EULAs as a means to avoid legal consequences.

Even though this is a large data set, gathering it revealed that of the 7,041 applications gathered, we were only able to extract 2,771 EULAs, i.e. around 39%. This can mostly be attributed to the number of extractors implemented and can thus be corrected by implementing more extractors. However, another contributing factor could be that some applications will not contain a EULA, but as the goal of this is detecting malicious software that uses EULA, this can be considered acceptable.

5.6.2 Proposed System Vulnerabilities

Our proposed automated system classifies EULAs as belonging to either malicious or benign programs, and then presents the results to the user. If the classified EULA is previously unknown by the system it could be considered that the system ask the user if the EULA content, together with meta-data about the associated binary program, could be collected. With this information it is then possible to automatically retrain the classifier with the new and slightly extended data set, i.e. using online learning. The alternative approach is offline learning, which involves collecting previously unseen EULAs by other means, and thereafter manually regenerating a new classifier at certain time-intervals, e.g. once a month. Regardless which method is being used the classifier would still automatically detect any attempt by the developers of malicious programs to fool the system by reformulating the content in their EULAs. The reason for this is that they always need to express their software’s behavior in the text, and by doing so they distinguish their EULA content from the benign EULAs.

However, it could be argued that the overall classification performance probably would be slightly higher if an online learning approach is used instead of an offline, since the classifier would be continuously
re-trained using new EULAs. Of course, a prerequisite for the online learning approach is that the integrity of the new EULAs and the associated meta-data can be guaranteed. Otherwise, it could be possible for external parties to feed the learning component with false EULA content and meta-data in an attempt to reach a certain conclusion for a specific EULA. The fundamental problem in such a scenario is whether the input from the clients to the server really can be trusted to be untampered with. Using cryptographic techniques it could be set up so that the data could not be modified in transit, but unfortunately it is harder to protect the client software from any unauthorized tampering.

5.6.3 Experimental Results

Our results show that both Bagging and the Random Forest algorithms handles the EULA categorization problem well, which is surprising as SVM often is considered the algorithm most suited for text categorization problems. Both Random Forest and Bagging outperforms SVM as can be seen in Figure 5.2. Naive Bayes Multinomial was also out performed since it showed quite poor performance results except when around 50 % of the features were reduced.

As shown in Figure 5.2, both Bagging and Random Forest show equivalent results when their default configurations were used, with a slight peak when 10 % of the features were reduced using CPD. Parameter tuning for both algorithms showed only slight alterations in performance. The results also indicate that configuring the Random Forest algorithm using 14 trees and 14 attributes present the best result within the context of EULA classification. For the Bagging algorithm the use of a bag size of 95 % and 14 iterations result in best performance. The Bagging algorithm reached the highest performance with an AUC measure of 0.997 and a standard deviation of 0.03, together with a low FNR of 0.062. The latter is important as trust in the proposed system otherwise could be lost if the proposed system suggests to the user that he/she should install a malicious application. From a user perspective
5. Informed Software Installation through License Agreement Categorization

it is less critical if the system suggest that the user shouldn’t install a legitimate program. It should be noted, that when novel instances, from outside our data set, are applied to the classifier, a certain performance degradation is expected.

5.7 Conclusion and Future Work

We have implemented and presented an automated tool for classification of binaries, based on the bundled EULA. We have created an algorithm, also presented in the paper, which handles the extraction. As the number of malicious programs increases, the presented system could assist users in separating between malicious and benign programs based on their EULA.

Furthermore, we have, as a result of using our automated tool created a dataset consisting of 2,771 EULAs. To the best of our knowledge, this is the largest collection of labeled EULAs available today. Using this dataset, we investigate the performance of four different learning algorithms, strongly suggesting the suitability of using Bagging and Random Forest to classify EULAs. The compilation and use of this extended dataset compared to previous datasets used in our prior experiments is a major contribution in this paper.

Using this dataset, we have investigated whether or not performance tuning of the learning algorithms provide better results than the standard settings. The results makes us conclude that the use of performance tuning is of limited use for the problem at hand. Similarly, we investigated the impact of using the feature selection algorithm CPD, which in other settings have proven very effective for increasing the prediction of the learning algorithm. However, we’ve found that, excluding Naive Bayes Multinomial, prediction is of almost no difference or even worse. In the case of Naive Bayes Multinomial, CPD increased the prediction, but otherwise performed worse than the other algorithms evaluated.
5.7. Conclusion and Future Work

For future work we plan to carry out experiments where computer users evaluate the use and benefit of a fully automated decision support tool when installing software. We will also investigate the occurrence of EULAs in a real world setting.
Social Network-based E-mail Classification

Anton Borg, Niklas Lavesson


Abstract

A majority of E-mail is suspected to be spam. Traditional spam detection fails to differentiate between user needs and evolving social relationships. Online Social Networks (OSNs) contain more and more social information, contributed by users. OSN information may be used to improve spam detection. This paper presents a method that can use several social networks for detecting spam and a set of metrics for representing OSN data. The paper investigates the impact of using social network data extracted from an E-mail corpus to improve spam detection. The social data model is compared to traditional spam data models by generating and evaluating classifiers from both model types. The results show that accurate spam detectors can be generated from the low-dimensional social data model alone, however, spam detectors generated from combinations of the traditional and social models were more accurate than the detectors generated from either model in isolation.
6. Social Network-based E-mail Classification

6.1 Introduction

The occurrence of spam has grown rapidly and it has been suggested that the majority of all E-mails are spam [81]. This development has resulted in the widespread use of spam filters, a use which can also be attributed to the inability of the current legislation to make an impact [82]. The legal inability has mainly been due to the differences in jurisdiction of various countries. Since most spammers only stay online for a limited amount of time it is considered hard to enforce the legislations, which increase the importance of automatic spam detection techniques [83].

This paper presents a method for E-mail spam detection that uses social information. This Online Social Network (OSN) supported spam detection method is compared with traditional spam detection. The paper also contributes with three metrics that have been adapted for social network data.

6.1.1 Aim and Scope

The aim is to investigate a method for spam classification using multiple OSN supported decision models. This paper implements a detection method based on using data from one OSN and compares it with a traditional spam detection method. The scope is limited to the study of social relationships mined from a public E-mail corpus.

6.1.2 Outline

Section 6.4 presents a method for E-mail spam detection using social information. In Section 6.3 research done in behavioral spam detection and in extension, OSN supported spam detection, is discussed. Section 6.5 details the method used for OSN data extraction, as well as the OSN data metrics. Section 6.6 outlines the experimental procedure. The
results are presented in Section 6.7 and discussed in Section 6.8. Finally, conclusions and future work is presented in Section 6.9.

6.2 Background

Internet users today use a number of media to share information. Communication media comprise SMS, MMS, OSNs, E-mail, and instant messaging services. These services contribute to information overload as a result of the amount of data presented to users via them. It has been stated that in 2010, around 250 billion E-mails were sent each day\(^1\). As much as about 90% of the E-mails sent are suspected to be spam\(^1\). E-mail is used along with OSNs as the two main forms of communication today. Large OSNs can attract around 100 million users, with Facebook surpassing 900 million\(^2\). People use OSNs to exchange messages, media and information concerning social activities.

E-mail and OSNs are rarely linked today. As such, the services are unable to use information from each other. An example is that a medium, e.g. E-mail, can use information provided by a second medium, e.g. OSNs, to combat the problem of information overload. Some work is focused towards this area, but are still in the initial phase [84], [85].

E-mail overload can be considered a specific form of information overload, a user receives more E-mails than he can process. Woods et al. [86] have found that people tend to characterize information overload in three different ways. These three ways are listed below, with descriptions of how they apply to the problem of E-mail overload.

\textbf{Clutter} is when there is too much information on the screen. A pro-

\footnotesize
\(^1\)How many emails are sent everyday, http://email.about.com/od/emailtrivia/f/emails_per_day.htm, 2012-02-26
posed solution is to remove data available, Woods et al argues that the removing agent still have to know which data to remove, stopping this solution from being ideal [86].

**Workload Bottleneck** occurs when a user is unable to properly deal with all the messages available within a timespan. Solutions are to have systems that summarize or prioritize the messages.

**Significance of data** concerns how to recognize which E-mails are important in a certain context. Some suggested solutions to this is, e.g. cognitive buoyancy, i.e. relevant information floating to the top, or message constellations, i.e. how a set of message relates to each other [87].

One attempt at addressing E-mail overload, is the improvement of spam detection.

### 6.3 Related Work

A number of reviews on the existing anti-spam techniques jointly conclude that automatic techniques are necessary to implement spam filtering [88] [89].

Some approaches based on the use of ontologies to classify E-mails based on content and previous messages, have aimed at generating personalized classifiers [90] [91]. Over time users will have gathered large amounts of E-mails. By constructing a profile based on E-mail habits, it is possible to detect outliers, i.e. spam [92]. Other research have investigated profiling a user’s E-mail sending behaviors using histograms to detect outliers [93] [94].

What can be inferred from previous work is a tendency towards using data from OSNs as a basis for anti-spam techniques. Researchers have previously investigated the use of OSN-based techniques for E-mail classification by using previous E-mail conversations to create a
correspondence graph, and from that graph, creating a model for classification [50] [53]. Most of the research so far has focused on building social networks from data, e.g. graph analysis, instead of gathering data from OSNs.

Learning algorithms have been investigated for prioritizing messages by building OSN from previous E-mail conversations [50]. The data represent messages submitted by volunteers. The results of the study show the feasibility of the approach. Two caveats with the study are that the data sets cannot be considered representative and the training of the model is irregular. The problem of representation occurs as the voluntarily submitted messages have been screen and selected by submitter. The second problem is that the training is done on the same amount of messages, regardless of the size of the data set. The amount used in the training set is the least common denominator for the data sets, i.e. a data set with a size greater than 1,000 instances will still use the same size of the training set as a data set with a size of 200. No practical reasons for this are mentioned.

Tran et al. have conducted research on providing a social context to E-mail correspondence [84]. A system that calculates the trust of the social path and also visualizes the path, have been implemented. This system provides an trust estimate between the corresponding parties. The data are based on social relationships from OSNs, in this case Facebook. OSN-based techniques can be used to enable the creation of a personalized spam filter and also allows the prioritization of messages, something which have been initially investigated [85].

By using the methods used to mine E-mail-based OSN and instead use other OSN sources as the basis of the classification, it is possible to address the problem of having a large E-mail based history, thus enabling extended classification for new users as well, given that said information is available on other OSN. Using OSN data sources as a complement to mining OSN data from E-mail corpora, removes the requirement of users having a large E-mail corpus to mine from, to be
The proposed method uses Social Network data sources in order to personalize and improve the classification of incoming messages. However, whether this approach is capable of detecting spam messages and which data is necessary needs to be investigated further.

### 6.4 Theoretical Model

Workload Bottleneck and Significance of data can be considered to be closely related. By solving Significance of data the likelihood of Workload Bottleneck can be reduced. By using automatic tools to determine cognitive buoyancy and message constellations, the E-mail overload can be re-
duced by classifying and prioritizing the messages. However, the question of which data to use as a basis for making these decisions is relevant. As users, in various contexts, use E-mails for different reasons, each user has to create personalized, context-aware classifiers. A classifier is an application that assigns labels to, in this problem domain, an E-mail, e.g. spam or ham.

The personalized and context-aware classifier uses, as a basis for its decision, several data sources that can be linked to the user. By using data available from different data sources, a classifier is able to interpret content and header information in a message and compare it with how a user communicates using the various data sources.

### 6.4.1 Data Sources

In this section a method is proposed that is capable of leveraging information from one medium of communication against a message received on another medium. As such the method needs to be able to gather information from several different data sources. These sources can be various services that a user has been linked to, e.g. various OSNs or E-mail history. The use of multiple data sources forms a classification basis that can be considered more personalized. For example, the content of the E-mail could be matched against a user’s profile information or against the corresponding party’s profile information, as well as earlier messages exchanged via the OSN. E-mail header information could be used to check whether a connection exists to a certain company or person via OSNs.

### 6.4.2 Context-driven Classification

The purpose and nature of social networks may vary. Some are used as a way of communicating short messages, some as a way of keeping
in touch with friends, some for professional relationships. As a result it is possible to use these social networks to distinguish between contexts. If context is taken into account, the importance can be estimated based on where the user is, what time it is and/or a specified user mode (e.g. work mode).

6.4.3 Knowledge-based Classification

Another aspect that can be taken into consideration is the level of knowledge of the contacts. By using OSN data, one can extrapolate, using e.g. work information or group memberships, a user’s knowledge area. Given such an approach, messages could be tagged as more relevant or less relevant depending on the perceived knowledge held by the author. One field of application where this aspect is useful to consider would be E-mail conversations involving multiple correspondents where the user want the most interesting reply in the thread to be the first read, for example replies in an list discussion.

6.4.4 Automatic E-mail Classification

E-mail classification can be done automatically. Let $I$ represent a set of E-mails represented as feature vectors. Each E-mail can be transformed into a vector of word frequencies. Let $C = \{\text{spam}, \text{ham}\}$ represent possible classifications. Given a set of examples, represented as pairs $E = \{<i, c>| i \in I, c \in C\}$, it is possible to generate an approximation, $\hat{f}$ of $f: I \rightarrow C$ using a supervised learning algorithm that generalizes from $E$.

Let $T$ be a similar example set, $T \cap E = \emptyset$. It is now possible to estimate the performance on $f$ by evaluating the performance on $T$. 

64
6.5 Method

The method is based on the idea of using several data sources as input to an engine that classifies a message as either spam or ham. These data sources could comprise pieces of information from several social media. Given data from these data sources, the engine creates a graph of users and extracts the social information. This social information is then used as a basis for the classification of incoming messages, regardless of which medium is used to transfer the message. The proposed method can be seen in Fig. 6.1.

This paper focuses on one particular data source. The proposed engine uses a supervised learning algorithm to generate spam detection models from both E-mail content, header data, and, social information. Using a model that is extended with OSN data that can help in determining the behavior of a user (for example, the relationship between the sender and receiver) can be regarded as an OSN-based model.

6.5.1 Social Data Generation

As no public E-mail corpora explicitly include social meta data, e.g., the explicit relationship between the sender and receiver, models are generated from existing E-mail headers. Thus, even if there is a lack of explicit OSN attributes in the data, it is possible to extrapolate certain social information from the data set.

The motivation for extracting OSN data from the E-mail corpus instead of using OSN data as a data source, is that previous research on OSN based classification has used private data sets which have been altered in undisclosed ways. A public data set that has been peer-reviewed has been chosen for use. As such, it is hard to link users in the data set to OSN profiles and extract OSN data, requiring the social information to be mined from the E-mail corpus.
6.5.2 Social Data Metrics

In order to add social information to the data set, data from the corpus is mined and social information is constructed. This paper focuses on three social attributes; the number of messages exchanged between users, the number of common contacts, and the number of participants. These metrics have been adapted from available OSN metrics. The number of exchanged messages indicates whether two users can be considered friends. Equation 6.1 describes the process of calculating the message-exchange score (MES) for a set of users associated with a message.

\[ \text{MES}(m) = \frac{\sum_{i=1}^{n} M_{s,t_i} + M_{t_i,s}}{n}. \]  

Equation 6.1

For a given message, \( m \), which contains a sender (s) and a set of receiving users (t), the number of messages to and from each user (\( t_i \)) and s is counted and an average for the number of receiving users (\( n \)) is calculated. In Equation 6.1, \( M \) is a matrix containing the number of messages between users.

Common Contacts Score (CCS) groups users, see Equation 6.2 It is calculated by counting the users that \( t_x \) and \( t_y \) both have exchanged bidirectional messages with.

\[ \text{CCS}(t_x) = \frac{\sum_{i=1}^{n} |A \cap B|}{n}. \]  

Equation 6.2

For a given user, \( t_x \), let set \( A \) contain the users that \( t_x \) have exchanged messages with, given that said exchange of messages is two-way. Let the same be true for set \( B \) for user \( t_i \). The cardinality of the intersection between \( A \) and \( B \) is used as the CCS. If this is done for several users (\( t_i \)), summarize the score and divide it by the number of participants (\( n \)) to get a mean.

The number of participants is equivalent to \( n + 1 \), as it includes the sender as well. These three attributes are added as separate attributes to the combined data set.
6.6 Experiments

The aim is to evaluate the classification performance impact of social data in comparison to traditional content-based spam detection, i.e. a multi-source model to a traditional. For this purpose, a public E-mail corpus is used to generate a social, combined and traditional data set. The social data set contains only the social information extracted from the E-mail corpus. The social extended set contains the message body, the available E-mail headers as well as the social information. The traditional data set contains only the headers and the message body, and uses no social data. The Weka machine learning workbench version 3.7.0 is used as the software platform for conducting the experiment [95].

6.6.1 Data Collection

The selected corpus is the TREC 2007 Public Corpus(Trec07). The Trec07 corpus was selected on the basis of the size as well as the feature set. Compared to other public domain corpora, such as the Enron Spam Corpus, Trec07 contains header data in addition to the content data. The Trec07 was collected in 2007 and the corpus consists of 25,220 ham and 50,199 spam.

6.6.2 Data Preprocessing

On a conceptual level, the data extracted from Trec07 can be divided into the list of attributes displayed in Table 6.1. However, few supervised learning algorithms can process strings. Thus, the string attributes must be transformed to a suitable representation. A common data model

---


6. Social Network-based E-mail Classification

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Classification: spam or ham</td>
<td>Nominal</td>
</tr>
<tr>
<td>To</td>
<td>Recipients</td>
<td>String</td>
</tr>
<tr>
<td>From</td>
<td>Sender</td>
<td>String</td>
</tr>
<tr>
<td>MES</td>
<td>Message-Exchange Score*</td>
<td>Numeric</td>
</tr>
<tr>
<td>NP</td>
<td>Number of participants*</td>
<td>Numeric</td>
</tr>
<tr>
<td>CCS</td>
<td>Common Contact Score*</td>
<td>Numeric</td>
</tr>
<tr>
<td>Received from</td>
<td>E-mail route description</td>
<td>String</td>
</tr>
<tr>
<td>Other headers</td>
<td>The remaining headers</td>
<td>String</td>
</tr>
<tr>
<td>Content</td>
<td>E-mail body, including any attachments</td>
<td>String</td>
</tr>
</tbody>
</table>

*Attributes only available in the Social and the Combined Data set.

Table 6.1: Attributes extracted from the Trec07 corpus

for this purpose, which has proven to be at least as effective as more complex solutions, is the bag-of-words model. In this model, strings are tokenized to words and represented by word vectors. In the first step, all special characters are removed, i.e., only the plus and minus signs, comma symbol, colon symbol, full stop symbol, the white space characters, alphabetical and numeric characters, as well as the @ sign are kept. The special characters preserved, are required to tokenize the E-mail header data and the whitespace is used to tokenize text strings into words.

In Weka, this transformation is carried out with the StringToWord-Vector filter, which is applied to the To, From, Received from, Other headers and Content attributes. The following filter configuration is employed: a maximum of 2,000 words are stored per category, term frequency-inverse document frequency (TF-IDF) is used for word frequency calculation, and the Iterated Lovins stemmer is used to reduce the number of words by keeping only the stems of words.

Artifacts in the data set among the attributes have been identified and removed. The X-headers are not included in the data set as X-headers can be considered artifact attributes. The artifact attributes have been added by software and have low predictive power.
6.6. Experiments

A stratified sample of 10% of the original instances is then generated, which leaves 7,541 instances. This is a size chosen to ensure a representative and large enough sample size, while maintaining a reasonable trade-off between computational effort and generalizability of the results.

6.6.3 Feature Selection

Categorical proportional difference (CPD), has been shown to outperform traditional feature selection methods, such as $\chi^2$, information gain, and odds ratio on several text categorization corpora [49]. Thus, CPD seems to be a suitable feature selection technique for the task at hand. The search for a suitable cutoff point for CPD is computationally expensive due to the possible non-linearity of the function of the number of kept words and the resulting performance [49]. A keep ratio interval is therefore defined and selected with a reasonable step size. In the presented study, a keep ratio interval of 1.0 to 0.5 together with a step size of 0.1 is used. This configuration yields 5 iterations for each data set, which lets the possible performance gain for each data set be determined, by keeping from 50% to 100% of the attributes.

6.6.4 Algorithm Selection

The main objective is to compare different data models available for detecting spam, hence the comparison of multiple learning algorithms to determine the optimal algorithm is out of scope. The Support Vector Machine (SVM) is a reasonable candidate, since it has been shown to work well with similar data models [96].

Given a set of examples, $E$, a SVM model, $\hat{f}$, is generated by mapping each example, $e \in E$, as a point on a plane [11]. The SVM model uses a kernel function for mapping the examples, enabling the instances to
be separated per class by a hyperplane. The hyperplane with the largest margin between the points of the classes is often chosen. Class prediction, $P$, for instances in $T$ are a result of which side of the hyperplane they are mapped to. These steps can be seen in Fig. 6.2. SVM’s predicted class is either 0 or 1 and as such, prediction probabilities are distorted. For this paper SVM, as implemented in the SMO algorithm available in Weka, is chosen with the default values used. To produce proper predictions, i.e. prediction probabilities between 0 and 1, `buildLogisticModels` is set to true.
6.6.5 Performance Evaluation

The true positive rate (TPR) and false positive rate (FPR) is defined as follows. False Positive (FP), True Positive (TP), False Negative (FN) and True Negative (TN) is used in the definitions.

\[
TPR = \frac{TP}{TP + FN}. \tag{6.3}
\]

\[
FPR = \frac{FP}{FP + TN}. \tag{6.4}
\]

Given a binary classifier, the Receiver Operating Characteristic (ROC) is the plot of the TPR versus the FPR, on the y-axis and x-axis respectively, for a set of instances, \( T \). In the domain of machine learning, given \( T \) and a corresponding set of predictions, each prediction is plotted with a one step distance relative to the previous point [11]. If \( p \in P \) is equal to \( c \) the instance is plotted along the y-axis, otherwise along the x-axis.

The area under the ROC curve (AUC) single point measure is used for evaluating the classifier performance, consisting of the portion beneath the ROC curve of the plot area. The larger portion of the plot area, i.e. higher AUC, denotes a higher performance. The AUC does not depend on an equal class distribution and misclassification cost [97]. In this paper, the weighted AUC (the average AUC of the classes) is used as a single point measure.

6.7 Results

Table 6.3 shows that, even though the number of attributes are lower, the classifier is still capable of producing good results. Compared to the traditional model, the results in Table 6.3 show a lower FPR for the combined model.
6. **Social Network-based E-mail Classification**

<table>
<thead>
<tr>
<th>Model</th>
<th>TPR (STD)</th>
<th>FPR (STD)</th>
<th>AUC (STD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social</td>
<td>0.953 (0.010)</td>
<td>0.380 (0.032)</td>
<td>0.926 (0.011)</td>
</tr>
<tr>
<td>Combined</td>
<td>0.992 (0.004)</td>
<td>0.000 (0.001)</td>
<td>0.999 (0.001)</td>
</tr>
<tr>
<td>Traditional</td>
<td>0.990 (0.004)</td>
<td>0.002 (0.003)</td>
<td>0.999 (0.001)</td>
</tr>
</tbody>
</table>

The results of the performance of SMO on the different models.

Table 6.2: Data model comparison

Table 6.2 shows a comparison between the different data models, with the advantage of showing OSN metrics as a individual data model. While the social data model produces a weighted AUC of 0.926, it has still got quite a high FPR of 0.380.

The performance of the different models are shown in Fig. 6.3, depicting the weighted AUC. The performance of the social model shows the feasibility of using the metrics and method suggested. To improve the model, the FPR need to be decreased. While the traditional and Combined model have high AUCs, this is most likely due to the time span that the messages were collected. A longer collection time should result in a lower score, as similarities between messages are fewer.

Nevertheless, the FPR of the combined model is lower than the traditional model. To determine the statistical significance for the differences between models, a larger amount of data is needed.

6.8 **Discussion**

Many of the OSN based techniques can be used in conjunction with traditional spam filtering techniques, to reduce the number of E-mails that need to be analyzed by the traditional technique. As such, even though a user cannot be linked to OSN, message can still be classified by traditional means.
A point that can be made is that a traditional, e.g. a Naive Bayes-based, spam classifier on a single users computer, given time and feedback, will have evolved into a personalized spam classifier. However, OSN-based classifiers do not require the same time span in order to become personalized since OSN-based classifiers use OSN data to bootstrap the feedback.
6. Social Network-based E-mail Classification

<table>
<thead>
<tr>
<th>Model (CPD*)</th>
<th>TPR (STD)</th>
<th>FPR (STD)</th>
<th>FNR (STD)</th>
<th>AUC (STD)</th>
<th>Nr. Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined (0.5)</td>
<td>0.998 (0.003)</td>
<td>0.026 (0.016)</td>
<td>0.002 (0.003)</td>
<td>0.995 (0.004)</td>
<td>1445</td>
</tr>
<tr>
<td>Traditional (0.5)</td>
<td>0.989 (0.011)</td>
<td>0.007 (0.005)</td>
<td>0.011 (0.011)</td>
<td>0.999 (0.002)</td>
<td>1442</td>
</tr>
<tr>
<td>Combined (0.6)</td>
<td>0.993 (0.012)</td>
<td>0.015 (0.017)</td>
<td>0.007 (0.012)</td>
<td>0.996 (0.003)</td>
<td>1734</td>
</tr>
<tr>
<td>Traditional (0.6)</td>
<td>0.981 (0.016)</td>
<td>0.004 (0.003)</td>
<td>0.019 (0.016)</td>
<td>0.997 (0.003)</td>
<td>1730</td>
</tr>
<tr>
<td>Combined (0.7)</td>
<td>0.968 (0.010)</td>
<td>0.000 (0.001)</td>
<td>0.032 (0.010)</td>
<td>0.998 (0.001)</td>
<td>2023</td>
</tr>
<tr>
<td>Traditional (0.7)</td>
<td>0.969 (0.010)</td>
<td>0.003 (0.003)</td>
<td>0.031 (0.010)</td>
<td>0.998 (0.001)</td>
<td>2018</td>
</tr>
<tr>
<td>Combined (0.8)</td>
<td>0.990 (0.005)</td>
<td>0.001 (0.002)</td>
<td>0.010 (0.005)</td>
<td>0.999 (0.001)</td>
<td>2311</td>
</tr>
<tr>
<td>Traditional (0.8)</td>
<td>0.988 (0.005)</td>
<td>0.003 (0.003)</td>
<td>0.012 (0.005)</td>
<td>0.999 (0.001)</td>
<td>2307</td>
</tr>
<tr>
<td>Combined (0.9)</td>
<td>0.992 (0.004)</td>
<td>0.001 (0.001)</td>
<td>0.008 (0.004)</td>
<td>0.999 (0.001)</td>
<td>2600</td>
</tr>
<tr>
<td>Traditional (0.9)</td>
<td>0.990 (0.004)</td>
<td>0.002 (0.003)</td>
<td>0.010 (0.004)</td>
<td>0.999 (0.001)</td>
<td>2595</td>
</tr>
<tr>
<td>Combined (1.0)</td>
<td>0.992 (0.004)</td>
<td>0.000 (0.001)</td>
<td>0.008 (0.004)</td>
<td>0.999 (0.001)</td>
<td>2888</td>
</tr>
<tr>
<td>Traditional (1.0)</td>
<td>0.990 (0.004)</td>
<td>0.002 (0.003)</td>
<td>0.010 (0.004)</td>
<td>0.999 (0.001)</td>
<td>2882</td>
</tr>
</tbody>
</table>

* The cut off point for the feature selection algorithm.

Table 6.3: Feature selection impact

6.8.1 Social Network Information

The methods for socially aware classifications are promising, but most of the research has been done by creating Social Networks from the E-mail corpus. While the method has been successful and shows feasibility, it requires a large E-mail corpus. For Social Network data to be extracted and built from the E-mail corpus requires a large E-mail corpus for there to be enough data available.

Social information could be extended using OSN data. For example, given that a user is active on an OSN, extracting and incorporating social information, similar to data that was used in the experiments, can be done. The information available on such networks has the potential to be of significantly larger quantities. Given that an E-mail classifier gain access to a users OSN data, that data could help to classify E-mails with none or only a few E-mail messages to aid in the classification process. Social Information-based classification could be used to prioritize messages, addressing the clutter and Significance of data characteristics of information overload. A consequence of this type of anti-spam filter, would be that bypassing the filter would require the spammers to
personalize their spam to an infeasible extent.

6.9 Conclusion and Future Work

This paper investigates the impact of spam classification based on social network data. The results suggest that accurate spam detectors can be generated from the low-dimensional social data model alone, however, spam detectors generated from combinations of the traditional and social models were more accurate than the detectors generated from either model in isolation. A theoretical model using several social network sources is presented. The social network metrics presented and used are adoptions meant to provide a normalized value for data extracted from various social networks. The performance of the social model suggests that the theoretical method presented merits further investigation.

For future work, a data set consisting of a larger number of messages that can be linked to various OSN needs to be created. Given such a data set, investigating the use of OSN data sources may yield more reliable results. The generalizability of the approach in this paper should be investigated on other data sets to verify the results found.
Abstract

Clustering algorithms have been used to divide genes into groups according to the degree of their expression similarity. Such a grouping may suggest that the respective genes are correlated and/or co-regulated, and subsequently indicates that the genes could possibly share a common biological role. In this paper, four clustering algorithms are investigated: k-means, cut-clustering, spectral and expectation-maximization. The algorithms are benchmarked against each other. The performance of the four clustering algorithms is studied on time series expression data using Dynamic Time Warping distance in order to measure similarity between gene expression profiles. Four different cluster validation measures are used to evaluate the clustering algorithms: Connectivity and Silhouette Index for estimating the quality of clusters, Jaccard
7. Comparison of Clustering Approaches for Gene Expression Data

Index for evaluating the stability of a cluster method and Rand Index for assessing the accuracy. The obtained results are analyzed by Friedman's test and the Nemenyi post-hoc test. K-means is demonstrated to be significantly better than the spectral clustering algorithm under the Silhouette and Rand validation indices.

7.1 Introduction

Gene expression microarrays are the most commonly available source of high-throughput biological data. Each microarray experiment is supposed to measure the gene expression levels of a set of genes in a number of different experimental conditions or time points. Improvements in gene analysis technology have resulted in an increase in gene expression data sets available, in both size and numbers [56, 98]. Gene clustering is one of the most important microarray analysis tasks when it comes to extracting meaningful information from expression profiles. For example, a high similarity between gene expression profiles may suggest that the corresponding genes are correlated and/or co-regulated, and subsequently indicates that the genes could possibly share a common biological role. Therefore, the ability to group genes according to their expression similarity can increase understanding of gene functions, cellular processes, and relationships between genes [54, 56, 57].

Several clustering algorithms have already been developed. However, there is no consensus on how to choose the appropriate clustering algorithm for partitioning gene expression profiles [56]. The choice of clustering algorithm is dependent on the data that is explored [54, 57]. The quality of the generated clustering solutions is also affected by the measure used to evaluate the similarity (distance) between gene expression profiles [54, 57].

In this paper, we investigate four clustering algorithms to partitioning of DNA microarray data. The algorithms are evaluated and benchmarked against each other on gene expression time series obtained from
7.2 Related Work

a study examining the global cell-cycle control of gene expression in fission yeast *Schizosaccharomyces pombe*. Time series expression profiles are expected to vary not only in terms of expression amplitudes, but also in terms of time progression since biological processes may unfold with different rates in response to different experimental conditions or within different organisms and individuals. Therefore, any classical distance metric as Euclidean, Manhattan etc. will produce a poor similarity score. In view of this, the studied algorithms use the Dynamic Time Warping distance in order to measure similarity between gene expression profiles. Four cluster validation measures are used to assess the performance of the algorithms, allowing evaluation and comparison of different aspects of the generated clustering solutions. Furthermore, statistical tests are applied to determine significant differences between algorithms.

The rest of this paper is organized as follows. Section 7.2 presents the related work. Section 7.3 describes the studied clustering methods. Section 7.4 introduces the experimental setup. Section 7.5 shows the validation results and present discussions. Section 7.6 concludes the paper.

### 7.2 Related Work

Clustering techniques have been one of the methods investigated to identify patterns of gene expressions, with the purpose of allowing an increased understanding of the function of gene expressions or relationships between gene expressions [54, 55]. These patterns can be detected based on the similarity, or distance, between gene expression profile pairs [99]. Two of the more common distance functions used are Euclidean distance and Pearson’s correlation [54]. Euclidean distance is not always preferable distance function when comparing gene expression time series as gene expressions series could have developed at different speeds. To compensate for differences in development speed, Dynamic
Time Warping (DTW) Distance can be used [100].

The choice of clustering technique is dependent on the data investigated and the distance metric applied [54, 56, 57]. More popular clustering techniques investigated have been k-means clustering, Self-Organized Maps (SOM) and hierarchical clustering [54, 56, 57]. Graph-based clustering and spectral-based clustering techniques have been suggested as having better performance [55, 57, 58, 98]. Among graph-based clustering algorithms, the cut-clustering algorithm has recently received attention [12, 101, 102]. Different evaluations of algorithms, however, have applied different cluster evaluation metrics on different data sets. The use of different metrics and data sets make comparing evaluations of algorithms non-trivial.

7.3 Methods

In this section the benchmarked algorithms are presented, followed by an explanation of each clustering algorithm. Four clustering algorithms are evaluated and benchmarked against each other: cut-clustering, k-means, spectral, and expectation-maximization (EM) algorithm. The algorithms are chosen either because of they are widely used or improved performance when clustering gene expression profiles has been suggested.

7.3.1 Cut-clustering Algorithm

The cut-clustering algorithm is a graph-based clustering algorithm, based on minimum cut tree algorithms to cluster the input data [101]. The input data is represented by a similarity (an undirected) graph, where each node is a data point and two nodes are connected if the similarity between the corresponding data points is positive, and the edge is weighted by the corresponding similarity score. The algorithm works
7.3. Methods

by adding an artificial node to the existing graph and connected to all nodes in the graph with a value \( \alpha \). A minimum cut tree is computed and the artificial node removed. The clusters consist of the nodes connected after the artificial node has been removed. The algorithm was implemented according to the specification [101]. A high \( \alpha \) value, results in a higher number of clusters produced, and vice versa. The \( \alpha \) value producing a specific number of clusters can be found using a binary search approach.

**Minimum Cut Tree**

A minimum cut tree is a tree that represents the minimum cut between pairs of nodes in a graph [103]. The parent of each node cut-clustering in the tree is tracked. Initially all the nodes in the graph are pointed to the first node. In each iteration, the source node, \( s \), is picked such that it has not been used before and the target node, \( t \), is the parent of the \( s \) node. Using the maximum flow algorithm, the minimum cut is then found between \( s \) and \( t \). Any neighbour belonging to \( t \) that is on the same side of the cut as \( s \) and have not been used as source, have their parent changed to \( s \). The maximum-flow algorithm is used \( n - 1 \) times.

**Adjacency Matrix**

An adjacency matrix, or similarity matrix, is a \( n \times n \) matrix-based representation of a graph where the value of \( A_{ij} \) represents the weight of the edge between the two nodes \( i \) and \( j \). This edge weight between nodes can be based on distance or similarity between nodes [12]. DTW can be used to compute the alignment between two time series, and an aligned distance score, termed DTW Distance, can be computed [100].
7. Comparison of Clustering Approaches for Gene Expression Data

7.3.2 K-means Clustering Algorithm

The k-means algorithm [104] is one of the most widely used techniques for clustering. It starts by initializing the $k$ cluster centers, where $k$ is determined prior to clustering. Then, each object (input vector) of the data set is assigned to the cluster whose center is the nearest. The mean (centroid) of each cluster is then computed so as to update the cluster center. This update occurs as a result of the change in the membership of each cluster. The process of re-assigning the objects and the update of the cluster centers is repeated until no more change in the value of any of the cluster centers. The implementation used is the SimpleKMeans available in the Weka framework [105]. To determine the distance between objects, the DTW Distance is used.

7.3.3 Spectral Clustering Algorithm

Spectral clustering has been found to generally detect good clustering solutions [12]. Spectral clustering is a graph-based clustering algorithm [12,106]. A Laplacian matrix, or similarity matrix, is computed. The eigenvector associated with the second-smallest eigenvalue of the matrix is computed and used to bipartition the graph [106]. This process can be repeated for the sub graphs, as deemed necessary. The implementation used is the spectral clusterer algorithm for Weka\(^1\). Similar to the k-means and cut-clustering algorithm, the spectral clustering algorithm uses DTW Distance as a distance measure.

7.3.4 Expectation-maximisation Clustering Algorithm

The EM algorithm is a distribution-based clustering algorithm. Distribution-based clustering algorithms assume that objects are created according to

\(^1\)http://www.luigidragone.com/software/spectral-clusterer-for-weka/
a probability distribution [107]. Different clusters can be considered created according to different probability distributions. For each object, the maximum likelihood of an object belonging to a specific cluster is computed. The number of clusters, $k$, is determined prior to clustering. EM is considered one of the most popular distribution-based clustering algorithms [107]. The Weka implementation of EM is used.

7.4 Experimental Setup

In this section, we first describe the microarray datasets used to demonstrate and evaluate the clustering algorithms described in Section 7.3. Then, we provide a short overview of the used cluster validation measures.

7.4.1 Microarray Datasets

The clustering results of the studied clustering algorithm are evaluated on gene expression time series data obtained from a study examining the global cell-cycle control of gene expression in fission yeast Schizosaccharomyces pombe [108]. The study includes eight independent time-course experiments synchronized respectively by: 1) elutriation (three independent biological repeats); 2) cdc25 block-release (two independent biological repeats, of which one in two dye-swapped technical replicates, and one experiment in a sep1 mutant background); and 3) a combination of both methods (elutriation and cdc25 block-release as well as elutriation and cdc10 block-release). Thus, nine different expression test sets are available: elu1, elu2, elu3, cdc25-1, cdc25-2.1, cdc25-2.2, cdc25-sep1, elu-cdc10 and elu-cdc25. In the pre-processing phase the rows with more than 25% missing entries have been filtered out from each expression matrix and any other missing expression entries have been imputed by
the DTWimpute algorithm [109]. In this way nine complete matrices have been obtained.

Rustici et al. identified 407 genes as cell-cycle regulated [108]. These have been subjected to clustering which resulted in the formation of 4 separate clusters. The genes that are not presented in the intersection of the nine original data sets have been removed. The latter produces a subset of 267 genes. Subsequently, the time expression profiles of these genes have been extracted from the complete data matrices and thus nine new matrices, which form our benchmark datasets, have been constructed.

The benchmark datasets have been additionally normalized by applying a data transformation method proposed [110].

### 7.4.2 Cluster Validation Measures

One of the most important issues in cluster analysis is the validation of clustering results. Essentially, the cluster validation techniques are designed to find the partitioning that best fits the underlying data, and should therefore be regarded as a key tool in the interpretation of clustering results.

Since none of the clustering algorithms performs uniformly best under all scenarios, it is not reliable to use a single cluster validation measure, but instead to use at least two that reflect different aspects of a partitioning. In this sense, we have implemented four different validation measures. To estimate the quality of clusters we use the Connectivity for assessing connectedness [111] and the Silhouette Index for assessing compactness and separation properties of a partitioning [112]. Further, the Jaccard Index is used for evaluating the stability of a clustering method [113]. The considered clustering method is randomized, such that, when applied \( p \) times, it produces different clustering results. We compute the averaged Jaccard Index over all \( p(p - 1)/2 \) pairs of \( p \)}
outcomes, assessing by this the stability of the method. Finally, we use the *Rand Index* for assessing accuracy [114]. This measure is applied to calculate the agreement between the clustering results generated by the considered clustering method and the known clustering solution (correct answers) [108].

**Connectivity**

Connectivity captures the degree to which genes are connected within a cluster by keeping track of whether the neighboring genes are put into the same cluster [111]. Let us define $m_{i(j)}$ as the $j$th nearest neighbour of gene $i$, and let $\chi_{im_{i(j)}}$ be zero if $i$ and $j$ are in the same cluster and $1/j$ otherwise. Then for a particular clustering solution (partition) $P = \{C_1, C_2, \ldots, C_k\}$ of matrix $M$, which contains the expression values of $m$ genes (rows) in $n$ different experimental conditions or time points (columns), the Connectivity is defined as

$$Conn(P) = \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{im_{i(j)}}.$$ 

The Connectivity has a value between zero and infinity and should be minimized.

**Silhouette Index**

Silhouette Index (SI) reflects the compactness and separation of clusters [112]. Suppose $P = \{C_1, C_2, \ldots, C_k\}$ is a clustering solution (partition) of matrix $M$, which contains the expression profiles of $m$ genes. Then the *Silhouette Index* is defined as

$$s(P) = \frac{1}{m} \sum_{i=1}^{m} \frac{b_i - a_i}{\max\{a_i, b_i\}},$$
7. Comparison of Clustering Approaches for Gene Expression Data

where \( a_i \) represents the average distance of gene \( i \) to the other genes of the cluster to which the gene is assigned, and \( b_i \) represents the minimum of the average distances of gene \( i \) to genes of the other clusters.

The values of Silhouette Index vary from -1 to 1 and higher value indicates better clustering results.

Jaccard Index

The Jaccard Index is used to evaluate the stability of a clustering method [113]. Given a pair of clustering solutions of the same data set (gene expression matrix \( M \)), \( P_1 \) and \( P_2 \), we define \( a \) as the number of gene expression profile pairs that belong to the same cluster in \( P_1 \) as well as in \( P_2 \). Let \( b \) be the number of gene expression profile pairs that belong to the same cluster in \( P_1 \) but not in \( P_2 \). Further, \( c \) is defined to be the number of gene expression profile pairs that belong to the same cluster in \( P_2 \) but not in \( P_1 \). The Jaccard Index between \( P_1 \) and \( P_2 \) is then defined as:

\[
J(P_1, P_2) = \frac{a}{a + b + c}.
\]

The Jaccard Index ranges from 0 to 1, where a higher value indicates a higher similarity between cluster solutions.

Rand Index

The Rand Index is used to calculate the accuracy. This allows for a measure of agreement between two clustering solutions (partitions), \( P_1 \) and \( P_2 \), of the same data set (gene expression matrix \( M \)). Each partition is viewed as a collection of \( m(m - 1)/2 \) pairwise decisions, where \( m \) is the number of genes. For each pair of genes \( g_i \) and \( g_j \) in \( M \), the partition either assigns them to the same cluster or to different clusters. Let \( a \) be the number of decisions where \( g_i \) is in the same cluster as \( g_j \) in \( P_1 \) and in \( P_2 \). Let \( b \) be the number of decisions where the two genes are
7.5. Validation Results and Discussion

placed in different clusters in both partitions. Total agreement can then be calculated using

\[
\text{Rand}(P_1, P_2) = \frac{a + b}{m(m - 1)/2}.
\]

The Rand Index ranges from 0 to 1, where a higher value indicates a higher accuracy.

7.5 Validation Results and Discussion

In this section, the performance of the clustering algorithms is studied on the benchmark gene expression matrices by using the cluster validation measures described in Section 7.4.2. Friedman’s test is further applied to discover significant differences between the algorithms.

Friedman’s test is a non-parametric statistical test that ranks the algorithms on data sets [115]. The average rank is computed for algorithms and used for comparison. Friedman’s test has been suggested as preferable when comparing algorithms over several data sets and normal distribution cannot be assumed [116]. If a significant difference has been detected by Friedman’s test, the Nemenyi test is used as post-hoc test. The Nemenyi test is used to detect between which algorithms significant differences exist. In the Nemenyi test a Critical Difference (CD) is computed. Algorithms where the difference between the average ranks exceeds the CD are considered to be significantly different.

7.5.1 Clustering Quality

In this section, we evaluate and compare the quality of the clustering solutions generated by the studied clustering algorithms discussed in Section 7.3 on the benchmark datasets described in Section 7.4.2 by using two cluster validation measures: Silhouette Index and Connectivity.
### 7. Comparison of Clustering Approaches for Gene Expression Data

Table 7.1: Average clustering measurement and average algorithm rank.

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>133</td>
<td>159.6</td>
<td>186.2</td>
<td>266</td>
<td>266</td>
<td>266</td>
<td>199.6</td>
<td>266</td>
<td>239.8</td>
<td>2.44</td>
</tr>
<tr>
<td>Spectral</td>
<td>66.8</td>
<td>25</td>
<td>45.5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>14</td>
<td>12</td>
<td>1.44</td>
<td></td>
</tr>
<tr>
<td>K-mean</td>
<td>199.2</td>
<td>162.8</td>
<td>183.7</td>
<td>177.8</td>
<td>172.2</td>
<td>182.7</td>
<td>155.9</td>
<td>104</td>
<td>162.9</td>
<td>3.00</td>
</tr>
<tr>
<td>EM</td>
<td>165</td>
<td>163</td>
<td>169.7</td>
<td>148</td>
<td>109.5</td>
<td>120.4</td>
<td>173.2</td>
<td>70</td>
<td>114</td>
<td>3.11</td>
</tr>
</tbody>
</table>


**Ra**: Average Rank

#### (a) Connectivity

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>-0.4851</td>
<td>-0.2920</td>
<td>-0.2053</td>
<td>0.8315</td>
<td>0.8385</td>
<td>0.9499</td>
<td>-0.4784</td>
<td>0.5799</td>
<td>0.6220</td>
<td>2.33</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.0098</td>
<td>0.0210</td>
<td>0.0063</td>
<td>-0.0782</td>
<td>-0.0758</td>
<td>-0.1135</td>
<td>-0.0547</td>
<td>0.0624</td>
<td>-0.0989</td>
<td>3.66</td>
</tr>
<tr>
<td>K-mean</td>
<td>0.5504</td>
<td>0.5393</td>
<td>0.4441</td>
<td>0.4723</td>
<td>0.3232</td>
<td>0.4242</td>
<td>0.3609</td>
<td>0.6456</td>
<td>0.5747</td>
<td>1.66</td>
</tr>
<tr>
<td>EM</td>
<td>0.5353</td>
<td>0.5112</td>
<td>0.3785</td>
<td>0.4222</td>
<td>0.3935</td>
<td>0.4023</td>
<td>0.3811</td>
<td>0.5627</td>
<td>0.5477</td>
<td>2.33</td>
</tr>
</tbody>
</table>


**Ra**: Average Rank

#### (b) Silhouette Index

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>0.9882</td>
<td>0.9881</td>
<td>0.7925</td>
<td>0.6455</td>
<td>0.6451</td>
<td>0.6458</td>
<td>0.6899</td>
<td>0.5348</td>
<td>0.4343</td>
<td>2.88</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.7354</td>
<td>0.7008</td>
<td>0.7228</td>
<td>0.9157</td>
<td>0.9426</td>
<td>0.9419</td>
<td>0.9299</td>
<td>0.8182</td>
<td>0.8388</td>
<td>2.33</td>
</tr>
<tr>
<td>K-mean</td>
<td>0.7317</td>
<td>0.7871</td>
<td>0.7138</td>
<td>0.8591</td>
<td>0.4782</td>
<td>0.6547</td>
<td>0.7783</td>
<td>0.8087</td>
<td>0.9921</td>
<td>3.33</td>
</tr>
<tr>
<td>EM</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.8543</td>
<td>1.0000</td>
<td>0.9129</td>
<td>0.7066</td>
<td>0.8574</td>
<td>0.9994</td>
<td>1.0000</td>
<td>1.44</td>
</tr>
</tbody>
</table>


**Ra**: Average Rank

#### (c) Jaccard Index

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>0.3423</td>
<td>0.3434</td>
<td>0.3734</td>
<td>0.5991</td>
<td>0.5980</td>
<td>0.5980</td>
<td>0.3410</td>
<td>0.5664</td>
<td>0.5085</td>
<td>3.11</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.4268</td>
<td>0.4418</td>
<td>0.4327</td>
<td>0.3615</td>
<td>0.3515</td>
<td>0.3487</td>
<td>0.3584</td>
<td>0.3899</td>
<td>0.3803</td>
<td>3.55</td>
</tr>
<tr>
<td>K-mean</td>
<td>0.6195</td>
<td>0.5863</td>
<td>0.6168</td>
<td>0.6488</td>
<td>0.6235</td>
<td>0.6387</td>
<td>0.6119</td>
<td>0.5366</td>
<td>0.5967</td>
<td>1.77</td>
</tr>
<tr>
<td>EM</td>
<td>0.7513</td>
<td>0.7578</td>
<td>0.7452</td>
<td>0.7778</td>
<td>0.5882</td>
<td>0.6514</td>
<td>0.7096</td>
<td>0.4939</td>
<td>0.5662</td>
<td>1.55</td>
</tr>
</tbody>
</table>


**Ra**: Average Rank

#### (d) Rand Index

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>Ra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>0.3423</td>
<td>0.3434</td>
<td>0.3734</td>
<td>0.5991</td>
<td>0.5980</td>
<td>0.5980</td>
<td>0.3410</td>
<td>0.5664</td>
<td>0.5085</td>
<td>3.11</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.4268</td>
<td>0.4418</td>
<td>0.4327</td>
<td>0.3615</td>
<td>0.3515</td>
<td>0.3487</td>
<td>0.3584</td>
<td>0.3899</td>
<td>0.3803</td>
<td>3.55</td>
</tr>
<tr>
<td>K-mean</td>
<td>0.6195</td>
<td>0.5863</td>
<td>0.6168</td>
<td>0.6488</td>
<td>0.6235</td>
<td>0.6387</td>
<td>0.6119</td>
<td>0.5366</td>
<td>0.5967</td>
<td>1.77</td>
</tr>
<tr>
<td>EM</td>
<td>0.7513</td>
<td>0.7578</td>
<td>0.7452</td>
<td>0.7778</td>
<td>0.5882</td>
<td>0.6514</td>
<td>0.7096</td>
<td>0.4939</td>
<td>0.5662</td>
<td>1.55</td>
</tr>
</tbody>
</table>


**Ra**: Average Rank
In Table 7.1a, the rows show the average Connectivity values generated on the clustering solutions obtained by the different clustering algorithms over the benchmark data sets. The average rank of the algorithms suggests that the spectral clustering algorithm performs best among the evaluated algorithms. Friedman’s test shows that there are some significant differences between the algorithms, \( X^2 = 9.4, \, df = 3, \, p = 0.05 \).

The Nemenyi test result (see Table 7.2a) shows that the spectral algorithm performs significantly better than the EM algorithm for the Connectivity measure. Although, there is a difference among the other algorithms, it is not significant.

Similarly, Table 7.1b shows the average Silhouette Index scores. The SI values of cut-clustering are varying between elutriation (elu1, elu2, elu3) and cdc25 block-release (cdc25-1,cdc25-2.1, cdc25-2.2) synchronization methods. There is no logical explanation of this and it is not observed for the other three algorithms. It may be due to the different number of clusters produced between the two methods. The average rank of the algorithms suggests that the k-means algorithm has the best performance.

Friedman’s test shows that there are some significant differences between the algorithms, \( X^2 = 11.4, \, df = 3, \, p = 0.01 \). The Nemenyi test result (see Table 7.2b) demonstrates that the k-means algorithm performs significantly better than the spectral clustering algorithm at a confidence level of both 0.95 and 0.99. Similarly to the Connectivity analysis, any significant differences between the other algorithms cannot be determined.

### 7.5.2 Clustering Stability

In this section, we evaluate and compare the stability of the generated clustering solutions using the Jaccard Index.
7. Comparison of Clustering Approaches for Gene Expression Data

Table 7.2: Paired rank comparison of algorithms. Upper triangle shows difference between algorithms. Lower triangle shows pairs with statistical significance.

<table>
<thead>
<tr>
<th></th>
<th>Cut-clustering</th>
<th>Spectral</th>
<th>K-means</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>1.000</td>
<td>0.556</td>
<td>0.667</td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>1.556</td>
<td>1.667</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td>0.111</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*: Significant at $p < 0.05$, CD: 1.563
**: Significant at $p < 0.01$, CD: 1.894

(a) Connectivity

<table>
<thead>
<tr>
<th></th>
<th>Cut-clustering</th>
<th>Spectral</th>
<th>K-means</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>1.333</td>
<td>0.667</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>2.000</td>
<td>1.333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
<td></td>
<td>* ', **</td>
</tr>
<tr>
<td>EM</td>
<td></td>
<td></td>
<td></td>
<td>0.667</td>
</tr>
</tbody>
</table>

*: Significant at $p < 0.05$, CD: 1.563
**: Significant at $p < 0.01$, CD: 1.894

(b) Silhouette Index

<table>
<thead>
<tr>
<th></th>
<th>Cut-clustering</th>
<th>Spectral</th>
<th>K-means</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>0.556</td>
<td>0.444</td>
<td>1.444</td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>1.000</td>
<td>0.889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
<td>1.889</td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

*: Significant at $p < 0.05$, CD: 1.563
**: Significant at $p < 0.01$, CD: 1.894

(c) Jaccard Index

<table>
<thead>
<tr>
<th></th>
<th>Cut-clustering</th>
<th>Spectral</th>
<th>K-means</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-clustering</td>
<td>0.444</td>
<td>1.333</td>
<td>1.556</td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>1.778</td>
<td>2.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
<td>0.222</td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td>*</td>
<td></td>
<td></td>
<td>' ', **</td>
</tr>
</tbody>
</table>

*: Significant at $p < 0.05$, CD: 1.563
**: Significant at $p < 0.01$, CD: 1.894

(d) Rand Index
7.5. Validation Results and Discussion

As one can notice the EM algorithm has the best performance under the Jaccard Index (see Table 7.1c). In general, the studied clustering algorithms have demonstrated a good stability. According to Friedman’s test, some significant differences exist between the algorithms, \(X^2 = 10.733, df = 3, p = 0.05\). The Nemenyi test result (see Table 7.2c) shows that only between one pair of algorithms can a significant difference be found. The EM algorithm performs significantly better than the k-means algorithm at a confidence level of 0.95.

7.5.3 Clustering Accuracy

In this section, the Rand Index is used to assess the accuracy of the clustering solutions generated by the studied clustering methods. For each generated clustering solution and the known clustering solution, a Rand Index score is calculated. The known clustering solution has been published [108].

The cut-clustering algorithm and spectral algorithm demonstrate comparatively low accuracy results (see Table 7.1d), \(Rand < 0.5\). The EM algorithm, however, has the best performance. Friedman’s test detect some significant differences between the algorithms, \(X^2 = 15.667, df = 3, p = 0.01\).

According to the Nemenyi test, a significant difference between algorithms can be found between two pairs of algorithms; the spectral and k-means algorithm, and the spectral and EM algorithm. The results (see Table 7.2d) show that the k-means algorithm performs significantly better than the spectral algorithm at a confidence level of 0.95. Further, the EM algorithm performs significantly better than the spectral algorithm at both 0.95 and 0.99 confidence levels. The difference between the other algorithms is not significant.
7. Comparison of Clustering Approaches for Gene Expression Data

7.6 Conclusion

In this work, four clustering algorithms have been studied: cut-clustering spectral, k-means and EM. Dynamic Time Warping distance has been applied in order to measure similarity between gene expression profiles. Four validation metrics have been used to assess cluster quality, stability and accuracy.

The k-means algorithm were found to be significantly better than the spectral algorithm for two cluster validation measurements, Silhouette Index and Rand Index. As such, given the results of the evaluation, the k-means algorithm should be the recommended clustering algorithm for this application over the spectral algorithm. It is interesting to note that for the graph-based algorithms, cut-clustering and spectral, a low Silhouette Index seem to correlate to a low Rand Index. This suggests that the Silhouette Index can be an indicator of the algorithms accuracy. In addition, the low Silhouette and Rand Indices scores may be due to the fact that the graph-based algorithms not always succeed in producing the right number of clusters, in comparison to the other two studied algorithms that determine the number of clusters prior to clustering.

The k-means algorithm is the recommended algorithm here, but algorithms should be properly evaluated when investigating new data before a choice is made. The cluster validation metrics used in this evaluation could be used or replaced with other cluster validation measurements. However, multiple cluster validation measurements should be used to increase the reliability of the evaluation. This can be noticed in Table 7.1, where the average rank of the algorithms varies between cluster validation measurement depending on the aspect measured.

Finally, it should be pointed out that the graph-based algorithms require tuning of the $\alpha$ value. The $\alpha$ value for producing the correct number of clusters can be found using a binary-search approach, but requires clustering solutions to be produced until the correct number of clusters is found.
Detecting Serial Residential Burglaries using Clustering

Anton Borg, Martin Boldt, Niklas Lavesson, Ulf Melander, Veselka Boeva


**Abstract**

According to the Swedish National Council for Crime Prevention, law enforcement agencies solved approximately three to five percent of the reported residential burglaries in 2012. Internationally, studies suggest that a large proportion of crimes are committed by a minority of offenders. Law enforcement agencies, consequently, are required to detect series of crimes, or linked crimes. Comparison of crime reports today is difficult as no systematic or structured way of reporting crimes exists, and no ability to search multiple crime reports exist.

This study presents a systematic data collection method for residential burglaries. A decision support system for comparing and analysing residential burglaries is also presented. The decision support system consists of an advanced search tool and a
plugin-based analytical framework. In order to find similar crimes, law enforcement officers have to review a large amount of crimes. The potential use of the cut-clustering algorithm to group crimes to reduce the amount of crimes to review for residential burglary analysis based on characteristics is investigated. The characteristics used are modus operandi, residential characteristics, stolen goods, spatial similarity, or temporal similarity.

Clustering quality is measured using the modularity index and accuracy is measured using the rand index. The clustering solution with the best quality performance score were residential characteristics, spatial proximity, and modus operandi, suggesting that the choice of which characteristic to use when grouping crimes can positively affect the end result. The results suggest that a high quality clustering solution performs significantly better than a random guesser. In terms of practical significance, the presented clustering approach is capable of reduce the amounts of cases to review while keeping most connected cases. While the approach might miss some connections, it is also capable of suggesting new connections. The results also suggest that while crime series clustering is feasible, further investigation is needed.
8.1 Introduction

Studies suggest that a large proportion of crimes are committed by a minority of offenders, e.g. in the USA, researchers suggest that 5% of offenders are involved in 30% of the convictions [42]. Law enforcement agencies, consequently, are required to detect series of crime, or linked crimes. A series can be defined as multiple offences committed by a serial offender. A serial offender can be defined as someone who has committed two or more crimes of the same type [37]. It is suggested by law enforcement in Sweden that, similarly to the international findings, a large proportion of the residential burglaries are committed by professional criminals that travel across large areas of Sweden. Simultaneously, according to the Swedish National Council for Crime Prevention, law enforcement agencies solved approximately three to five percent of the 21,300 reported residential burglaries in 2012.

The detection of linked crimes is helpful to law enforcement for several reasons. Firstly, the aggregation of information from crime scenes increases the amount of available evidence. Secondly, the joint investigation of multiple crimes enables a more efficient use of law enforcement resources [37].

Law enforcement needs to handle a large amount of reported crimes, and the detection of series of crimes are often carried out manually. A decision support system that enables law enforcement to decrease the amount of cases when reviewing crimes would increase resource efficiency.

Forensic evidence, e.g. DNA, and fingerprints, has been used to detect linked crimes [15, 42]. The availability of forensic evidence is, however, limited [42]. In the absence of forensic evidence, behavioural information can be used as an alternative data source [15]. A criminal committing a series of crimes has been found to have a high intra-crime behavioural similarity [37]. Similarly, behavioural consistency tends to be lower between criminals in similar situations [37].
This article presents a new decision support system (DSS) that can be used to systematically collect burglary data and to perform visualisations, analyses, and interpretations of the collected data. The article evaluates a key component of the DSS: the use of clustering techniques to group burglaries based on different definitions of similarity between burglaries, described in Figure 8.1. Clustering has been used to group data according to similarity between data points, or to find communities in the data. Clustering residential burglaries based on different similarity aspects would potentially allow law enforcement to find series whilst reviewing a smaller amount of residential burglaries, i.e. used as a case selection DSS. Consequently, the use of this DSS would allow law enforcement agencies to save resources, whilst providing individual investigators with increased support. The clustering is performed using the cut clustering algorithm [101].
8.2. Decision Support System for Residential Burglary Analysis

8.1.1 Purpose Statement

The purpose of this study is two-fold. First, a DSS for collecting, managing and analysing residential burglary information is presented. Secondly, the potential of minimum cut based graph clustering of crimes is investigated to reduce the amount of crimes to review to detect series of residential burglaries. The impact of different edge representations and edge removal criteria on cluster quality and accuracy is investigated. Clustering quality is measured using the modularity index and accuracy is evaluated by applying the rand index.

The data comprises residential burglary reports gathered from southern Sweden and the Stockholm area.

8.1.2 Outline

The remainder of this work is organized as follows: Section 8.2 presents a DSS for residential burglary analysis. In Section 8.3, the related work is reviewed. Section 8.4 then describes the minimum cut clustering algorithm. In Section 8.5 and Section 8.6, the methodology and experimental procedure is described. The results of the experiments are presented in Section 8.7 and analysed in Section 8.8. Conclusions and future work is presented in Section 8.9.

8.2 Decision Support System for Residential Burglary Analysis

Since 2011, researchers from Blekinge Institute of Technology collaborate with law enforcement officers and analysts from the Blekinge county police as well as four additional county police authorities from southern Sweden. The aim is to develop Information and Communication Tech-
8. Detecting Serial Residential Burglaries using Clustering

Technology (ICT) solutions for law enforcement. The scope is currently limited to solutions that target residential burglaries. The strategies, tactics, and overall organisational structure of the police vary between countries but the main issues are shared between many countries.

In Sweden, the police is organised into 21 county police authorities, or regional units, where each correspond to a particular county. The National Police Board (NPB) is the central administrative and supervisory authority of the police service. The NPB comprises The National Bureau of Investigation and the Swedish Security Service. In addition, the Swedish police includes the Swedish National Laboratory of Forensic Science. In 2015, the Swedish police will be re-organised into one national authority.

The collaboration between Blekinge Institute of Technology and the Swedish police was formed to improve the capability to solve residential burglary cases. In particular, the police are interested in ICT software, and organisational changes, that improve the data exchange and collaborative efforts of multiple county police authorities when addressing serial crime. Engineers and researchers at Blekinge Institute of Technology developed a prototype DSS for this purpose in 2012. Since then, the collaboration between academia and police has been extended to encompass authorities responsible for two thirds of the Swedish population.

The DSS uses a web-based graphical user interface, which is connected through program logic to a database with structured information about residential burglaries. The crime data is collected through a digital form, which can be observed in Appendix B, which is being continuously developed in close collaboration between Blekinge Institute of Technology and the Swedish police. The form forces police officers at the crime scene to acquire specific pieces of information about the modus operandi, the physical location, and other types of information related to each crime. Before the introduction of this form, the data collected varied extensively between crime scenes with respect to quality, amount, personal bias, and perspective.
The program logic in the DSS is centered around a straight-forward search engine interface, which makes it possible to search, filter, group, and compare crime scenes with respect to various properties related to modus operandi, location, and so on. This can be seen in Figure 8.2. In addition to the comprehensive search engine, the DSS features a plugin-based analysis framework, which makes it possible to develop specific types of descriptive and inferential statistical analyses of the crime scene data.

This article is focused on an analysis component developed for the DSS. The component makes it possible to perform clustering on crime scene data for various purposes. The aim of this article can therefore be described as two-fold: to introduce and describe the DSS and the structured data collection of crime scene data as well as to evaluate one
particular type of analysis component.

**8.3 Related Work**

The problem of linking reported crimes has mostly been investigated from a psychological or criminological perspective. The research has focused on crimes conducted that can be considered violent, e.g. sexual offences, rapes, homicides, and different types of burglaries, including violent burglaries [15,37,38,39,40,41].

The research conducted suggests that behavioural consistency is present among offenders and that there exists an inter-individual variation [37]. The behavioural consistency between similar situations tends to increase with the experience of the perpetrator. More specifically, an individual tend to behave similarly in similar situation. Multiple individuals tend to behave differently, to a certain degree, in similar situation [37]. The smaller temporal proximity between situations usually results in an increased similarity for a perpetrator.

Different aspects of behaviours can be used for comparison, e.g. modus operandi (MO), spatial proximity, and temporal proximity. The MO can be further divided into three domains; entry behaviour, target characteristics, and goods stolen [43]. Entry behaviour describes the procedure used to enter the premises, e.g. broke and entered through a window on the second floor. Target characteristics denote characteristics of the residence being targeted, i.e. isolated location, two-story building, alarm, etc. Recent research on using MO characteristics have suggested the effectiveness of the characteristics [37]. Spatial proximity has been shown to increase the hit ratio, i.e. the number of detected linked cases, for some crime types, e.g. burglaries [37]. Spatial proximity have also been investigated for use in groupings of crimes to detect where crimes concentrate in space and time, e.g. to detect hotspots, or to predict future crime locations [30,31,32,33,34,35]. Spatiotemporal correlations over
longer time periods have been investigated to further enhance hotspot
detection [36]. These approaches differs from crime linkage in that they
detect areas which are more likely to have crimes committed, whereas
crime linkage finds connections between crimes over larger areas [30].
Different hotspot methods are used in DSS for law enforcement agen-
cies, e.g. to detect areas for resource prioritization [34,35].

Some researchers have computed the similarity between pairs of
crimes based on various behaviours. Many of these studies have used
similarity coefficients between cases, such as the Jaccard coefficient, to
represent behavioural consistency [37]. The similarity scores have been
used as input for logistic regression analysis as well as used to plot a
receiver operating characteristics (ROC) curve for linked and unlinked
cases [38,41,42,43]. The results have suggested that spatial proximity,
and temporal proximity, are better indicators to determine linked crime
than the MO characteristics [41]. The MO characteristics, however, was
still found to be a significant indicator [41,43]. Using only temporal and
spatial proximity, a model was created which was able to correctly clas-
sify 86.9% of crime pairs in the sample, compared to 80% for a model
using spatial proximity and 75.6% for a model using temporal proxim-
ity [41]. The MO characteristics-based selection achieved an accuracy
between 54.4% and 58.1%.

The data used in many of the reviewed studies were extracted from
law enforcement agencies, in some cases according to a checklist [41].
Since the data extraction was done after the case information was re-
ported, the case information might be incomplete, as law enforcement
officers might not have reported crimes in a systematic way, e.g. differ-
ent aspects are considered important.

The overarching theme studied in the previous articles can be de-
scribed as detecting crimes that are similar. Detecting crimes that are sim-
ilar can be considered similar to the purpose of clustering, where the
goal is to distribute objects into separate groups.

Several investigations have tried to compute similarity scores be-
8. Detecting Serial Residential Burglaries using Clustering

tween pairs of crimes. Such scores can easily be translated into a graph structure or an adjacency matrix. A graph can be described as a set of nodes that can be connected with vertices of different weights, e.g. a set of crimes as nodes with a similarity score as vertex weight. A survey of the graph clustering domain was conducted in 2007 [12]. Graph clustering have successfully been used to identify communities/networks in other settings [117,118,119]. Community detection have been investigated extensively and different methods summarized [117]. A graph can be divided into clusters based on a split criterion. This approach is denoted divisive clustering. The split criterion can be computed using several methods, e.g. maximum-flow, spectral methods, and Markov chains [12].

Previous work on the problem of linking residential burglaries have suggested that there is a difference between the similarities of linked and unlinked residential burglaries. The difference have been investigated using pairs of crimes [38,41,42,43]. While the pair-wise comparison have suggested a possibility of detecting links between cases, the studies have not investigated approaches for detecting series of crimes. Consequently, each series of residential burglaries should have a high intra-series similarity score and a low inter-series similarity score, similar to the description of community detection in graphs [117]. As such, clustering residential burglaries can be described as a problem of grouping instances or detecting communities within the data. One of the more recent graph clustering algorithms suitable for detecting communities in graphs is the cut clustering algorithm suggested by Flake [101,102,117].

8.4 Cut Clustering Algorithm

The cut clustering algorithm is a graph-based clustering algorithm based on minimum cut tree algorithms to cluster the input data [101]. The input data used is an undirected graph where the edges between nodes could represent a similarity or distance measure.
8.4. Cut Clustering Algorithm

The algorithm can be described as follows: an artificial node is added to the existing graph and connected to all nodes in the graph with the edge value \( \alpha \). A tree is created from the graph using the minimum cut tree algorithm [103]. The artificial node is then removed from the tree and the nodes that are still connected are considered part of different clusters [101].

Algorithm 2 Cut Clustering Algorithm [101]. Input is a graph \((G)\) with nodes \((V)\) and edge weights \((E)\).

1: function \text{CutClustering}(G(V, E), \alpha)
2: \(V' \leftarrow V \cap t\)
3: for all nodes \(v \in V\) do
4: Connect \(t\) to \(v\) with edge weight \(\alpha\)
5: end for
6: \(G'(V', E')\) is the expanded graph after connecting \(t\) to \(V\)
7: Calculate the Min-cut Tree \(T'\) of \(G'\)
8: Remove \(t\) from \(T'\)
9: return all connected components as clusters of \(G\)
10: end function

The cut clustering algorithm (see Algorithm 2) is implemented according to the original description [101]. The minimum cut tree algorithm is implemented according to Gusfield’s specification [103]. Gusfield’s algorithm is described further in Section 8.4.2. To find the minimum cuts between two nodes in the adjacency matrix, the Edmond-Karp maximum flow algorithm is used.

A property of the maximum flow algorithm is that complete graphs, or near complete graphs, can result in trivial clustering solutions. This is due to that, in a complete graph, the minimum cut can be trivial, i.e. cutting either the source or target node. In such cases, the trees created will be either star-shaped, i.e. each node is connected directly to the root node, or unary, i.e. each parent containing one node. The clustering produced from such a tree will be trivial. Consequently, this needs to be considered when creating the graph.
8.4.1 The $\alpha$ Value

The $\alpha$ value is used when the artificial node is attached to the other nodes. The outcome of the minimum cut clustering algorithm is determined by the $\alpha$ value [101]. The behaviour of the $\alpha$ value can be predicted. Given a high $\alpha$ value, several clusters will be produced. As the $\alpha$ value decreases, fewer clusters will be produced.

The $\alpha$ value can, when the number of desired clusters is known, be discovered using e.g. a binary search for alpha values until the wanted number of clusters is found [101]. If the number of desired clusters is unknown, a binary search can iterate over the $\alpha$ value until trivial clusters are no longer produced or the number of clusters produced is stabilized. This has been implemented according to Algorithm 3 [120]. The boundary values are chosen so that the clustering solutions produced with the boundary values will be trivial, i.e. either a single cluster or several singletons.

8.4.2 Minimum Cut Tree

Minimum cut trees can be created using, for example, two well-known algorithms, the Gomorra-Hu algorithm or Gusfield’s algorithm [103]. In both, the maximum-flow algorithm is used $n - 1$ times. However, Gusfield’s algorithm is considered simpler in its implementation as the algorithm operates on a adjacency matrix as a representation and requires no contractions or expansions of the graph, contrary to the Gomorra-Hu algorithm. Parallel implementations are supported by both algorithms. Gusfield’s algorithm is presented in Algorithm 4.

In Gusfield’s algorithm the parent of each node in the tree is tracked. Initially all the nodes in the graph are pointed to the first node. In each iteration, the source node, $s$, is picked such that it has not been used before and the target node, $t$, is the parent of the $s$ node. Using the maximum flow algorithm, the minimum cut is then found between $s$
Algorithm 3 Binary search for iterating alpha values

1: min ← Min(G(E))
2: max ← Max(G(E))
3: Cl ← 1
4: Cr ← |V_G|
5: while min < max − 1 and Cl < Cr do
6:   alpha ← (min + max)/2
7:   c ← |CutClustering(G(V, E), alpha)| \(\triangleright\) c gets the number of clusters in the clustering
8:   if c = Cl then
9:     min ← alpha
10:  else if c = Cr then
11:    max ← alpha
12:  else
13:    if c > Cl and c < (Cr/2) then
14:      Cl ← c
15:    else if c < Cr and c > (Cr/2) then
16:      Cr ← c
17:    else
18:      End Loop
19:  end if
20: end if
21: end while

and t. Any neighbour belonging to t that is on the same side of the cut as s and have not been used as source, have their parent changed to s.

8.5 Data and Method

8.5.1 Data Collection

The data consist of residential burglary incident reports collected in a systematic way by law enforcement officers over a period of six months.
Algorithm 4 Gusfields Minimum Cut Tree Algorithm [103]

1: function MinCutTree(G(V, E), c) \Comment{A weighted, undirected graph}
2:  for i = 1 → |V_G| do
3:  \hspace{1em} tree_i ← 1
4:  end for
5:  for s = 2 → |V_G| do \Comment{|V_G| − 1 maximum flow iterations}
6:  \hspace{1em} t ← tree_s
7:  \hspace{1em} flow_s ← max-flow(s, t)
8:  \hspace{1em} \{X, \bar{X}\} ← minimum s-t cut
9:  \hspace{1em} for u ∈ V_G, u > s do
10:     \hspace{2em} if u ∈ X then
11:        \hspace{3em} tree_u ← s
12:     \hspace{2em} end if
13:  \hspace{1em} end for
14:  end for
15:  VT ← V_G \Comment{Build the minimum cut tree}
16:  ET ← ∅
17:  for s = 2 → |V_G| do
18:  \hspace{1em} ET ← ET \cup \{s, tree_s\}
19:  \hspace{1em} f(\{s, tree_s\}) ← flow_s
20:  end for
21:  return \(T = (V_T, E_T, f)\)
22: end function

The incident reports are collected through a checkbox-based form, providing a common base of data collected. The form used consists of eleven sections and 107 checkboxes. In addition to the checkboxes, information about time, date and geographical position (longitude, latitude and street address) of the reported incident is also gathered. If required, a field for unstructured textual descriptions or observations also exists. This field allows law enforcement officers to enter additional information of importance.

The incident reports have been gathered from the southern part of Sweden and the Stockholm area. The reports comprise 2,416 reported
residential burglaries. Of the incident reports, law enforcement officers have provided anonymized information about suspects for 24 residential burglaries, allowing connections between cases to be established.

8.5.2 Data Representation

The instances are inserted into an $n \times n$ adjacency matrix, and for each pair in the adjacency matrix a similarity index is computed as an edge representation. This process is repeated so that adjacency matrices exist for several similarity indices.

The produced clustering solutions are saved using the DIMACS format\textsuperscript{1}.

**Edge Representation**

The edges in the graph are represented by different similarity coefficients, making the edge weights a measure of similarity between nodes. The similarity coefficients have been chosen based on results suggested in previous research. First, the Jaccard index is computed between crime pairs based on three different MO characteristics, complete MO characteristics, residential characteristics, and stolen goods information. Secondly, spatial and temporal proximity is computed between crime pairs based on geodesic distance and temporal distance (measured in days) respectively. The Jaccard calculation is expanded upon in Appendix A.

**Edge Removal Criteria**

The minimum cut tree algorithm, when given complete graphs or near-complete graphs, can produce trees that are star-shaped, i.e. each node

\textsuperscript{1}http://lpsolve.sourceforge.net/5.5/DIMACS_maxf.htm, 2013-02-24
8. Detecting Serial Residential Burglaries using Clustering

is connected directly to the root node, or unary (see Section 8.4). Consequently, it is possible that the clustering can be improved by converting complete graphs into incomplete graphs. Two approaches for this conversion are investigated.

In the first approach, several threshold values are computed and the graphs are pruned based on these values, by keeping only the edges where the nodes are considered similar to a certain degree. Threshold edge removal for graph transformation can be considered a global approach, in that a single threshold value is computed and used for all edges in the graph. Only edges where the nodes are considered similar to a certain degree, e.g. below the threshold value, are kept. The thresholds are defined as the mean and the quartile values.

The second approach use time and distance based measures, and given the outcome the edge is either removed or the weight is changed to indicate lesser similarity. The distance-based edge removal can be considered local, i.e. only a single edge is investigated at a time. Given this, the criteria for removing an edge can be different for each edge. The measures are based on The Mantel Cross product adaption, and the Journey Time Distance (JTD) [35]. The JTD criteria removes edges between cases that are physically impossible to have been committed by the same burglars, i.e. the spatial distance is too large for the temporal span. The Mantel Cross product adaption is based on the Mantel index, which is a correlation test between time and distance for pairs of instances [121]. Both measures are expanded upon in Appendix A.

8.5.3 Cluster Validation Measurements

The following cluster validation measurements are used to measure the quality and accuracy of the minimum cut clustering algorithm.

True Positive (TP) is a pair of nodes in the same cluster that are linked to each other. False Negative (FN) is a pair of nodes in different clusters
that are linked to each other. *True Negative (TN)* is a pair of nodes in different clusters that are not linked to each other. *False positive (FP)* is a pair of nodes in the same cluster that are not linked to each other.

*Rand Index (RI)* is the percentage of correct decisions, i.e. how well the clustering algorithm has grouped the residential burglaries. RI for clustering can also be denoted *Accuracy*. One problem with RI is that, in certain cases, as the number of clusters increase, the RI increases [122]. The RI is computed as:

\[
RI = \frac{TN + TP}{TN + TP + FP + FN}
\]  

(8.1)

*Modularity* is a cluster quality index that can be used to measure how well the clusters group and separate instances, i.e. *intra-cluster density* and *inter-cluster sparsity*. It is based on the premise that the fraction of edges between nodes in a cluster should be higher than the expected fraction of edges between nodes in a cluster to indicate significant group structure, see Equation 8.2 [119, 123, 124]. The modularity index maps onto [-1,1].

\[
Q = \sum_{c \in C} \left[ \frac{|E(c)|}{|E|} - \left( \frac{\sum_{v \in c} \text{deg}(v)}{2 \times |E|} \right)^2 \right]
\]  

(8.2)

*Coverage* is a cluster quality index based on intra-cluster density. It is related to modularity, as modularity is in essence coverage subtracted with the expected coverage. Coverage computes the edges within a cluster divided by the total number of edges, see Equation 8.3.

\[
\text{Cov} = \sum_{c \in C} \frac{|E(c)|}{|E|}
\]  

(8.3)
8.6 **Experiment Design**

The following two aspects of residential burglary clustering are investigated. First, the impact of different similarity indices as edge representations and of different edge removal criteria on the quality of the clusters produced. Second, the performance with which the minimum cut algorithm is able to group residential burglaries without splitting series of crimes.

8.6.1 **Hypothesis**

The following hypotheses are investigated in this study.

**Experiment 1** The hypotheses of Experiment 1 can be described as follows:

The choice of edge representation and edge removal criteria can positively affect the quality of the clusters produced. If the null hypothesis is not supported, the alternate hypothesis states that the choice of edge representation affects the quality of the clustering.

**Experiment 2** The hypothesis of Experiment 2 is that high quality clustering solutions of residential burglaries can result in fewer crimes to analyze whilst keeping series intact.

8.6.2 **Experiment 1: Cluster Quality**

The first experiment investigates how different edge representations and edge removal criteria affect the quality of the clusters created by the minimum cut clustering algorithm. The experiment consists of two indepen-
dent variables: edge representation and edge removal criteria. Each variable has several levels as described in section 8.5.2 and 8.5.2. As such a $X \times Y$ factorial design, where $X$ and $Y$ corresponds to the variable levels, is used as an experimental design [125].

The dependent variable of the experiment design is the modularity. Each combination of variable levels is tested 10 times. For each repetition, a subsample of the dataset is created using simple random sampling with replacement. The subsample consists of 250 instances.

A between-subjects factorial analysis of variance (ANOVA) is used to evaluate a factorial experiment design. The between-subjects factorial analysis of variance allows evaluation of possible interaction between variables, as well as evaluating significant difference between variables and levels. Interaction is when the combination of two variables affect each other and thus the dependent variable in a unpredictable way [115].

If there is a significant difference between the factorial combinations, a post-hoc test is used after the between-subjects factorial analysis of variance to detect which factorial combination performs better. In this case, the post-hoc test used is Fisher’s LSD test [115]. Fisher’s LSD test is vulnerable to type II errors, i.e. incorrectly supporting a null hypothesis, but have a lower chance of making type I errors, i.e. incorrectly rejecting a null hypothesis. If the difference for a comparison is less than Fisher’s LSD value ($CD_{LSD}$), the null hypothesis is supported. The statistical tests are conducted using R and the ezAnova package.

### 8.6.3 Experiment 2: Crime Distinction

The second experiment investigates whether residential burglaries can be clustered with high quality whilst keeping series of crime intact. The accuracy of the clustering is measured using the RI. The experiment design is similar to Experiment 1 and consists of two independent variables: edge representation and edge removal criteria. Each variable has
8. Detecting Serial Residential Burglaries using Clustering

several levels as described in section 8.5.2 and 8.5.2. Similar to Experiment 1, $X \times Y$ factorial design, is used. The dependent variables of the experiment design are the modularity and RI.

This experiment uses the labeled instances. Labeled instances are instances where law enforcement agencies have provided information whether the instance is known to be part of a series or not. For each repetition, a subsample of the dataset is created using simple random sampling with replacement from the labeled instances. The subsample consists of 24 instances. The experiment uses an identical design to Experiment 1, with the exception of an additional dependent variable, RI.

For the second experiment, the statistical test outlined in Section 8.6.2 is carried for both dependent variables. To detect relationships between the modularity and RI, Pearson’s correlation coefficient is used.

8.7 Results

8.7.1 Experiment 1

According to the modularity cluster validation measure (see Table 8.1a), the 1st-Quantile has the worst performance of all the different edge removal criteria. Similarly, the edge Jaccard Goods and Temporal proximity representations are performing worse than other representations. The performances of these edge representations indicate that Jaccard Goods and Temporal proximity are unsuitable for representing differences between crime cases. The goods available in the form are a few general items, e.g. such as electronics.

It should be noted that none of the modularity results are positive, indicating that the cluster solutions produced have a lower fraction of edges within clusters than the expected fraction of edges between clus-
### Results

#### (a) Modularity

<table>
<thead>
<tr>
<th></th>
<th>1st-Quantile</th>
<th>2nd-Quantile</th>
<th>3rd-Quantile</th>
<th>Average</th>
<th>MeanTest</th>
<th>None</th>
<th>JTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>d2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>d3</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>d4</td>
<td>0.000</td>
<td>0.003</td>
<td>0.001</td>
<td>0.063</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.006)</td>
<td>(0.001)</td>
<td>(0.200)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>d5</td>
<td>0.000</td>
<td>0.016</td>
<td>0.004</td>
<td>0.018</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>(0)</td>
<td>(0.050)</td>
<td>(0.012)</td>
<td>(0.056)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
</tbody>
</table>

* d1-d5: Jaccard Goods, Jaccard Residence, Jaccard MO, Spatial Proximity, Temporal Proximity.
* Standard Deviation within parentheses.

#### (b) Coverage

<table>
<thead>
<tr>
<th></th>
<th>1st-Quantile</th>
<th>2nd-Quantile</th>
<th>3rd-Quantile</th>
<th>Average</th>
<th>MeanTest</th>
<th>None</th>
<th>JTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>180.8</td>
<td>147.6</td>
<td>250.0</td>
<td>142.0</td>
<td>164.2</td>
<td>165.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(250.0)</td>
<td>(134.8)</td>
<td>(122.5)</td>
<td>(53.3)</td>
<td>(117.7)</td>
<td>(171.2)</td>
<td></td>
</tr>
<tr>
<td>d2</td>
<td>136.1</td>
<td>152.8</td>
<td>114.6</td>
<td>135.2</td>
<td>126.9</td>
<td>159.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(250.0)</td>
<td>(86.1)</td>
<td>(149.6)</td>
<td>(93.9)</td>
<td>(135.9)</td>
<td>(156.8)</td>
<td></td>
</tr>
<tr>
<td>d3</td>
<td>132.5</td>
<td>181.9</td>
<td>193.5</td>
<td>141.6</td>
<td>136.0</td>
<td>186.2</td>
<td></td>
</tr>
</tbody>
</table>

* d1-d5: Jaccard Goods, Jaccard Residence, Jaccard MO, Spatial Proximity, Temporal Proximity.

#### (c) Number of Clusters

Table 8.1: Mean Clustering Measurement for Experiment 1.
8. Detecting Serial Residential Burglaries using Clustering

As such, the clustering solution can not be said to have separated clusters well. The clustering solution has most likely created to many clusters of crimes, meaning that crimes that are most likely supposed to be grouped together are not. This makes the job of the analyst harder in that the number of crimes is reduced too much and connections could be overlooked.

The coverage of several different edge representations and edge removal criteria scored 0, indicating that a high number of singleton clusters were produced (see Table 8.1b), i.e. that crimes are not considered to be connected to any other crimes. The coverage cluster validation score is not surprising the negative modularity score, as the modularity index incorporate similar measure similar aspects. However, as the coverage score is mostly 0, focus will be placed on the modularity score henceforth.

8.7.2 Experiment 2

In the results of the clustering solutions produced for Experiment 2, a pattern can be observed in that none of the clustering solutions yield a high modularity index (see Table 8.2a). The highest mean modularity index that can be observed is $-0.018$, followed by $-0.019$. As the modularity index is an index that ranges from $-1$ to $1$ and where a positive index value indicate a higher number of edges within the clusters, the indices produced cannot be considered good. In fact, all pairings of edge representations and edge removal criteria produce a negative modularity index. Of the edge removal criteria, the first quartile and the JTD function have the worst modularity score. Looking at the corresponding pairings in Table 8.2d, these edge removal criteria have produced clustering solutions that almost consist of singular clusters, i.e. the number of clusters is close to the number of nodes. Looking at the number of clusters produced (see Table 8.2d), several factor combinations produce a high number of singleton cluster solutions, i.e. groups of crimes can not be produced. The coverage cluster validation measurement (see Ta-
### 8.7. Results

Table 8.2: Mean Clustering Measurement for Experiment 2.

<table>
<thead>
<tr>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st-Quantile</td>
<td>2nd-Quantile</td>
<td>3rd-Quantile</td>
<td>Average</td>
<td>Mean</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.0107)</td>
<td>(0.060)</td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.024)</td>
<td>(0.075)</td>
<td>(0.092)</td>
<td>(0.099)</td>
<td>(0.052)</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.037)</td>
<td>(0.089)</td>
<td>(0.012)</td>
<td>(0.012)</td>
<td>(0.012)</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.068)</td>
<td>(0.207)</td>
<td>(0.094)</td>
<td>(0.095)</td>
<td>(0.103)</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.096)</td>
<td>(0.096)</td>
<td>(0.096)</td>
<td>(0.096)</td>
<td>(0.096)</td>
</tr>
<tr>
<td>0.250</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>(0.013)</td>
<td>(0.059)</td>
<td>(0.092)</td>
<td>(0.092)</td>
<td>(0.075)</td>
</tr>
</tbody>
</table>

* d1-d5: Jaccard Goods, Jaccard Residence, Jaccard MO, Spatial Proximity, Temporal Proximity.

* Standard Deviation within parentheses.
8. Detecting Serial Residential Burglaries using Clustering

Table 8.2: Mean Clustering Measurement for Experiment 2.

<table>
<thead>
<tr>
<th></th>
<th>1st-Quantile</th>
<th>2nd-Quantile</th>
<th>3rd-Quantile</th>
<th>Average</th>
<th>MeanTest</th>
<th>None</th>
<th>JTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>0.993</td>
<td>0.854</td>
<td>0.600</td>
<td>0.988</td>
<td>0.512</td>
<td>0.558</td>
<td>0.988</td>
</tr>
<tr>
<td></td>
<td>(0.006)</td>
<td>(0.144)</td>
<td>(0.346)</td>
<td>(0.014)</td>
<td>(0.348)</td>
<td>(0.309)</td>
<td>(0.006)</td>
</tr>
<tr>
<td>d2</td>
<td>0.986</td>
<td>0.706</td>
<td>0.634</td>
<td>0.690</td>
<td>0.464</td>
<td>0.605</td>
<td>0.986</td>
</tr>
<tr>
<td></td>
<td>(0.011)</td>
<td>(0.198)</td>
<td>(0.249)</td>
<td>(0.294)</td>
<td>(0.302)</td>
<td>(0.215)</td>
<td>(0.009)</td>
</tr>
<tr>
<td>d3</td>
<td>0.987</td>
<td>0.589</td>
<td>0.612</td>
<td>0.665</td>
<td>0.639</td>
<td>0.767</td>
<td>0.985</td>
</tr>
<tr>
<td></td>
<td>(0.011)</td>
<td>(0.255)</td>
<td>(0.244)</td>
<td>(0.282)</td>
<td>(0.103)</td>
<td>(0.148)</td>
<td>(0.010)</td>
</tr>
<tr>
<td>d4</td>
<td>0.984</td>
<td>0.535</td>
<td>0.809</td>
<td>0.522</td>
<td>0.619</td>
<td>0.759</td>
<td>0.984</td>
</tr>
<tr>
<td></td>
<td>(0.011)</td>
<td>(0.283)</td>
<td>(0.155)</td>
<td>(0.220)</td>
<td>(0.370)</td>
<td>(0.351)</td>
<td>(0.012)</td>
</tr>
<tr>
<td>d5</td>
<td>0.988</td>
<td>0.699</td>
<td>0.651</td>
<td>0.648</td>
<td>0.658</td>
<td>0.607</td>
<td>0.982</td>
</tr>
<tr>
<td></td>
<td>(0.011)</td>
<td>(0.276)</td>
<td>(0.263)</td>
<td>(0.306)</td>
<td>(0.301)</td>
<td>(0.259)</td>
<td>(0.013)</td>
</tr>
</tbody>
</table>

* d1-d5: Jaccard Goods, Jaccard Residence, Jaccard MO, Spatial Proximity, Temporal Proximity.

*(c) Rand Index

In Table 8.2c, for the groupings that have produced singular clustering solutions, a high rand index has been observed. This result however, can be attributed to the data, which in singular clusters, have several instances should not be connected and thus increase the rand index score. That is to say, when the algorithms creates groups of singular crimes, the fact that it successfully separates crimes that should not be connected takes precedence over connecting crimes that should be connected. But
the goal is to produce smaller groups of crimes for analysis, not single crimes. As such, when looking at the accuracy of the grouped crimes (i.e. rand index), one must also consider secondary cluster validation measurements, e.g. the number of clusters produced or the modularity index. A low number of clusters and a high rand index can be considered positive, as this would mean that the number of crimes to analyse is reduced. Similarly, a high modularity and a high rand index can also be considered indicative of a good clustering solution. A high number of clusters together with a high rand index indicates a clustering solution that have scattered known crimes, e.g. series of crimes are not grouped. An example of a clustering solution can be seen in Figure 8.3. The modularity and rand index of the cluster solution is $-0.0103$ and $0.5579$ respectively. In this example, the cut clustering algorithm has not been able to keep series intact, which is indicated in the low rand index.

Figure 8.3: Cluster Solution example for Spatial Proximity and 3rd-Quantile based on labeled data. Clusters are connected by edges.
8. Detecting Serial Residential Burglaries using Clustering

8.8 Analysis

The results of the experiments are analysed using an ANOVA test to detect if there exist any statistically significant difference between the variables. Fisher’s LSD test is used to detect between which variables statistical significant differences exist.

8.8.1 Experiment 1

The means and standard deviations are presented in Table 8.1. For the modularity cluster validation measurement, two factor analysis of variance showed a significant effect for the edge representation, $F(4, 315) = 6.112, p < .05$; a significant effect for the edge removal criteria, $F(6, 315) = 20.001, p < 0.005$; but not any significant effect for the interaction of the two factors, $F(24, 315) = 2.733$, non-significant. As such, there is a difference in the performance between the edge representation and between the edge removal criteria, but they do not affect each other.

For the edge representation, Fisher’s LSD post-hoc test found that the Jaccard Residence edge representation scored significantly better than both the Temporal proximity and Jaccard Goods representation (see Table 8.3a). Similarly, Spatial proximity and Jaccard MO scored significantly better than Jaccard Goods edge representation. This indicates that residential characteristics, followed by Spatial proximity, is to prefer when representing similarity or distance between crimes.

For the edge removal criteria, Fisher’s LSD test indicates that the MeanTest criterion performs significantly better than the other criteria. The best performance, however, was achieved when not applying any edge removal criterion. The 3rd-Quantile and 2nd-Quantile perform significantly better than the 1st-quantile. The poor performance of the 1st-Quantile can be attributed to the high number of edges being removed from the graph, with the consequence that the nodes cannot be
8.8. Analysis

Table 8.3: Fisher’s LSD post-hoc test for Experiment 1. The group column contains letters which denote group belonging and where different letters represent groups that are statistically significantly different from each other.

connected. As the performance of not applying an edge removal criterion was similar to the best performing edge removal criteria, there is little reason to apply an edge removal criterion. Using edge removal criteria would increase the number of computational steps required when grouping crimes.

8.8.2 Experiment 2

The means and standard deviations are presented in Table 8.2. For the modularity cluster validation measurement, two factor analysis of variance showed a significant effect for the edge representation, \( F(4, 315) = 4.339, p < .05 \); a significant effect for the edge removal criteria, \( F(6, 315) = 69.735, p < 0.001 \); and a significant effect for the interaction of the two factors, \( F(24, 315) = 2.733, p < 0.001 \). As such, there is a significant difference between edge removal criteria, between different edge representations, and the two variables affect each other.

Fisher’s LSD post-hoc test for the edge representation found that there is a difference between the Jaccard MO, Jaccard Residence, spatial
8. Detecting Serial Residential Burglaries using Clustering

<table>
<thead>
<tr>
<th>Edge Removal Criteria</th>
<th>Means</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd-Quantile</td>
<td>−0.0362</td>
<td>a</td>
</tr>
<tr>
<td>MeanTest</td>
<td>−0.0490</td>
<td>a</td>
</tr>
<tr>
<td>2nd-Quantile</td>
<td>−0.0570</td>
<td>a</td>
</tr>
<tr>
<td>None</td>
<td>−0.0656</td>
<td>ab</td>
</tr>
<tr>
<td>Average</td>
<td>−0.0881</td>
<td>b</td>
</tr>
<tr>
<td>JTD</td>
<td>−0.2279</td>
<td>c</td>
</tr>
<tr>
<td>1st-Quantile</td>
<td>−0.2500</td>
<td>c</td>
</tr>
</tbody>
</table>

\[ CD_{LSD}: 0.0298 \]

(a) Modularity

<table>
<thead>
<tr>
<th>Edge Removal Criteria</th>
<th>Means</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st-Quantile</td>
<td>0.9880</td>
<td>a</td>
</tr>
<tr>
<td>JTD</td>
<td>0.9854</td>
<td>a</td>
</tr>
<tr>
<td>Average</td>
<td>0.7032</td>
<td>b</td>
</tr>
<tr>
<td>2nd-Quantile</td>
<td>0.6771</td>
<td>b</td>
</tr>
<tr>
<td>3rd-Quantile</td>
<td>0.6617</td>
<td>bc</td>
</tr>
<tr>
<td>None</td>
<td>0.6595</td>
<td>bc</td>
</tr>
<tr>
<td>MeanTest</td>
<td>0.5788</td>
<td>c</td>
</tr>
</tbody>
</table>

\[ CD_{LSD}: 0.0875 \]

(b) Rand Index

Table 8.4: Fisher’s LSD post-hoc test for edge removal criteria. The group column contains letters which denote group belonging. Different letters represent groups that are statistically significantly different from each other.

proximity representations and the Jaccard Goods representation. This is similar to the results in Experiment 1, suggesting similarity of clustering ability over the data sets. For the edge removal criteria, the LSD test found significant difference between multiple criteria. The different groups can be seen in Table 8.4a. Criteria belonging to different groups are significantly different, with group a performing significantly better than group b. Group a performs significantly better than group b and c. Fisher’s LSD test found, for the interaction between the factors, that the Jaccard Residence with MeanTest criteria, Jaccard MO with MeanTest and 3rd-Quantile criteria, and spatial proximity with 3rd-Quantile criteria performed significantly better than factors paired with JTD and 1st-Quantile.

For RI, two factor analysis of variance did not show a significant effect for edge representation, \( F(4, 315) = 0.663 \), non-significant; a significant effect for edge removal criteria, \( F(6, 315) = 27.729, p < 0.05 \); and a significant effect for the interaction of the two factors, \( F(24, 315) = 2.198, p < 0.05 \). As such, the two variables interact and affect the performance of each other.

The results of Fisher’s LSD post-hoc test for the interaction between
### 8.8. Analysis

<table>
<thead>
<tr>
<th>Edge Representation</th>
<th>Edge Removal Criteria</th>
<th>Modularity</th>
<th>Rand Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Means</td>
<td>Group</td>
</tr>
<tr>
<td>Jaccard Residence</td>
<td>1st-Quantile</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>2nd-Quantile</td>
<td>−0.035</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>3rd-Quantile</td>
<td>−0.045</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>−0.070</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>MeanTest</td>
<td>−0.018</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>−0.032</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>JTD</td>
<td>−0.208</td>
<td>d</td>
</tr>
<tr>
<td>Jaccard MO</td>
<td>1st-Quantile</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>2nd-Quantile</td>
<td>0.037</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>3rd-Quantile</td>
<td>0.021</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.029</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>MeanTest</td>
<td>0.019</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.073</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>JTD</td>
<td>0.224</td>
<td>d</td>
</tr>
<tr>
<td>Jaccard Goods</td>
<td>1st-Quantile</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>2nd-Quantile</td>
<td>0.092</td>
<td>bc</td>
</tr>
<tr>
<td></td>
<td>3rd-Quantile</td>
<td>0.046</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>MeanTest</td>
<td>0.053</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.057</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>JTD</td>
<td>0.226</td>
<td>d</td>
</tr>
<tr>
<td>Spatial Proximity</td>
<td>1st-Quantile</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>2nd-Quantile</td>
<td>0.032</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>3rd-Quantile</td>
<td>0.021</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.030</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>MeanTest</td>
<td>0.081</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.121</td>
<td>c</td>
</tr>
<tr>
<td></td>
<td>JTD</td>
<td>0.235</td>
<td>d</td>
</tr>
<tr>
<td>Temporal Proximity</td>
<td>1st-Quantile</td>
<td>0.250</td>
<td>d</td>
</tr>
<tr>
<td></td>
<td>2nd-Quantile</td>
<td>0.089</td>
<td>bc</td>
</tr>
<tr>
<td></td>
<td>3rd-Quantile</td>
<td>0.048</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.060</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>MeanTest</td>
<td>0.072</td>
<td>abc</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.045</td>
<td>ab</td>
</tr>
<tr>
<td></td>
<td>JTD</td>
<td>0.246</td>
<td>d</td>
</tr>
</tbody>
</table>

Table 8.5: Fisher’s LSD post-hoc test for combinations of Edge representation and Edge Removal Criteria. The group column contains letters which denote group belonging. Different letters represent groups that are statistical significantly different from each other.
8. Detecting Serial Residential Burglaries using Clustering

the two factors, Edge Representation and Edge Removal Criteria, see Table 8.5. Interaction for both the Modularity and Rand Index is shown. Best modularity mean was found for combinations of Jaccard Residence or Jaccard MO edge representation together with Mean-Test edge removal criteria; Jaccard MO or Spatial Proximity edge representation together with 3rd-Quantile edge removal criteria. Together with the results of Experiment 1, this further supports the suggestion that crime similarity is best represented using Residence, MO or spatial proximity characteristics. The modularity cluster validation measurement results suggest that the use of JTD or 1st-Quantile edge removal criteria produces clustering solutions that are significantly worse than other factor combinations. This suggests the unsuitability of these edge removal criteria. The interaction result of Fisher’s LSD post-hoc test for the Rand Index validation measurement, found significant difference between several groups of edge removal criteria (see Table 8.4b). Similar to the modularity index, this suggests that certain characteristics are better representations when the goal is the accuracy of grouping crimes. The 1st-Quantile and JTD performed significantly better than the other criteria. Fisher’s LSD test for the interaction between the factors found that pairings with the JTD and 1st-Quantile performed significantly better than other pairings, except for Jaccard Goods with the 2nd-Quantile edge removal criteria and spatial proximity with the 3rd-Quantile edge removal criteria. The resulting clustering solutions, however, were trivial or near trivial, i.e. crimes were separated into singletons or too small groups. If the clustering solutions consist of too small groups, or singletons, using it as a selection system is inadvisable.

An interesting aspect to note is that there are factorial combinations that belong to high performing groups for both the Modularity and Rand Index. This would indicate that the clustering solutions provide non-trivial group separation with a above average accuracy and that these combinations are more suitable to crime separation than others. The Spatial Proximity edge representation and the 3rd-Quantile edge removal criteria is one example where a high cluster separation is found, as well as a good accuracy in assigning instances to clusters. Similar
can be said of the Jaccard MO edge representation and where no edge removal criteria have been applied (marked as None in Table 8.5).

The calculated Pearson’s correlation coefficient between the Rand Index and the Modularity cluster validation measurement is found to be $-0.768$. This can be seen in Figure 8.4. This indicates that for Experiment 2, the Modularity score of the cluster solutions decreases as the Rand Index increases, i.e. as crime groupings get worse the accuracy increases. It should be noted that this is for all different factor combinations, including ones creating trivial clustering solutions. A few outliers can be found with a Modularity score of 0.2 or higher and with a high Rand Index.

### 8.8.3 Validity Threats

Validity threats against this study are outlined and discussed in this section. It should be noted that over time, most of these validity threats will be corrected with the help of larger data sets and more labeled information about the data.

First, accuracy can be questioned as the number of labeled instances is quite small. Consequently, one cannot conclude strong generalization when it comes to accuracy. There is not much one can do about this, as the systematic gathering of crime reports in Sweden is still quite young and have not been adopted by all police counties. Also, the number of solved cases is quite low, three to five percent. As the amount of labeled data is quite small, it is quite possible that some cases should be part of a series but the information is not available yet.

Second, the crime reports have been gathered during a rather short time period of one year. It could be that during this time, crimes have occurred in a pattern that can be considered non-representative of criminals. Similarly can be said of the counties from which the crime reports have been collected. This is being rectified as data gathering takes place
8. Detecting Serial Residential Burglaries using Clustering

and more counties chooses to join the systematic process.

8.8.4 Discussion

The results of the experiments highlight two interesting things. First is the idea of edge removal criteria, which the results indicate have no benefit on the grouping of crimes. Consequently, when grouping residential burglaries there is little incentive to not have similarity information between cases that fulfill edge removal criteria. While the use of edge
removal criteria might have more impact on the cut-clustering adaption than for the domain, it also suggests that retaining information between cases is important. Initially, edge removal criteria such as Journey Time Distance would remove edges/connections between crimes that was not feasible to be connected. However, Journey Time Distance turned out to be one of the worst performing ways of removing edges between crimes, which is surprising. It, intuitively, seemed that by removing edges between cases that were impossible to connect, the clustering algorithm would perform better, which was not the case.

The second interesting outcome is the different suitability for representing similarity between crimes. The edge representations with the best performance were Residence characteristics, MO characteristics, and Spatial proximity. This is unsurprising, as similar results have been suggested in previous research conducted. The goods representation has the worst performance. This is most likely due to the fact that the goods attributes available are quite limited and certain items tend to be more attractive to criminals. That the data are gathered during a limited time period might also affect the impact of the goods. The high performance of the MO and Residence characteristics suggests that there exists a certain behavioural consistency within crimes, and that this consistency differ between criminals to such an extent that it is possible to differentiate.

Something to consider is that law enforcement officers are more likely to solve crimes committed by local criminals. Crimes committed by non-local criminals that are active over a larger area, during a limited time, are more complicated to solve. There might be a difference in which representation of crime similarities have the best performance between local and non-local criminals.

This also means that there will be an imbalance in the data, i.e. a series will not be a large part of the crime cases. Imbalance will always exist, as serial residential burglars do not conduct all burglaries and of 22000 residential burglaries a year, one group might only have commit-
8. Detecting Serial Residential Burglaries using Clustering

ted a minority. As can be observed in Figure 8.4, as the groupings get worse, the accuracy score is increased. This is mostly due to the correct classification of singleton cases outweighing the false classification of connected cases. Consequently, one of the problems is to find a balance between group size and series quality.

Given the nature of the area, there will be crimes that are singletons. These cases could be a singleton in the clustering solution, or be part of a larger group of crimes. As the purpose is to reduce the number of crimes an analyst has to analyse, as well as to suggest new connections among crimes, this is only a problem as long as clustering solutions are mostly singletons. This is not always the case (see Figure 8.3). It is possible that an increased performance from the cut-clustering algorithm, or another clustering algorithm, can create better clustering solutions. Similarly, the method might also fail to cluster cases that should be connected. As of today no clustering methods are employed for clustering crimes in Swedish law enforcement agencies and if a specific investigator does not have a notion of a connection between cases, they will be missed entirely. So while an approach such as this might miss a crime when grouping, the systematic selection of cases is still preferable as a complement to the expertise of a single investigator, who otherwise keep earlier cases and suspects in memory.

It should be noted that as the clustering uses distance-based metrics, e.g. Jaccard index, when the MO for a perpetrator changes (which it most likely will over time), the distance between newer and older crimes will increase. This could be problematic, and should be something to be kept in mind by investigators. A recommendation is that a time window for crimes to investigate is limited to 6 months, due to change in MO and other variables [126]. A similar issue is the MO is often a consequence of circumstances. A criminal known for a certain way of entering the building might, when encountering an unlocked door not use the same MO and as a result, that specific case will not be same MO. This is a consequence of only using a single MO characteristic as a basis for the distance measurement. A potential remedy to this would be to use a
measurement that combines multiple MO characteristics for calculating the distance between crimes. However, this problem is also mitigated by the fact that the implemented DSS has knowledge of physical evidence gathered at the scene. As such, the investigator might have a set of cases where shoe prints and fingerprints have been found, and can search for local cases with similar physical evidence and see if matches exist. Such an approach can detect other MO used by the same burglars.

8.9 Conclusion

In this article a DSS for managing and analysing systematically gathered residential burglary reports have been presented. The DSS allows law enforcement to easily search and compare residential burglary reports. The DSS contains, among other modules, an analytical framework. The use of clustering to group residential burglaries in the DSS has been investigated, using several similarity criteria.

While results of the modularity cluster validation measurement indicate that the separation between clusters is poor, the Rand Index results still supports further investigation into this area. The first experiment concerned which representation of residential burglaries for use with clustering, and whether edge removal methods increased performance. The results of the first experiment shows that the choice of edge representation, but not the edge removal criteria, positively affected the modularity score. The second experiment concerned the whether clustering solutions where able to correctly cluster crime series. The results in Experiment 2 suggest that, when excluding trivial clustering solutions, a high quality clustering solution results in with an above chance accuracy, i.e. $0.5 > RI > 0.8$. The experiments have suggested that the choice of which edge representation to use when grouping crimes can positively affect the clustering solution. Best performance is found using Spatial proximity or Residential characteristics as a basis for comparing crimes. As such, this would indicate that these characteristics are to prefer when
law enforcement investigates related residential burglaries.

The clustering solutions without any edge removal criteria performed within the same group as the highest scoring edge removal criteria in most cases. The mean modularity score of the experiments were, however, suggesting that the cut clustering algorithm is not optimal for this domain. An increase in the ability to correctly cluster crimes would allow law enforcement officers to investigate fewer amounts of criminal cases with a lower chance of missing cases in a series of crimes. The results suggest that while clustering crime series are feasible using cut clustering, further investigation is needed.

We have identified six aspects for future work. First, we only had access to a low number of labeled cases and the experiment needs to be investigated using a larger labeled data set. Getting access to a larger labeled data set is not trivial since identifying series of crimes is hard. Second, in this study only individual edge representations have been investigated. Researchers have found that in some cases, combinations of edge representation scores have had a better performance than the individual edge representation. A distance index using combined crime characteristics, such as spatial and MO characteristics, needs to be developed. Third, it should be noted that the performance of other clustering algorithms on this domain has not been investigated, and a comparison of algorithms should be conducted. Fourth, an accuracy index that takes into account the imbalance of the data would be better suited than the Rand Index.
Abstract

Studies suggest that a minority of offenders commits a large proportion of crimes. Automatic estimation of the likelihood that two crimes are related reduces the manual labor required by investigators. This article investigates the possibility to use logistic regression analysis to estimate the link probability. Previous research has suggested the feasibility of this approach. The logistic regression model was constructed using data collected from three police counties in Sweden and evaluated using a bootstrap approach. The model was evaluated against the expert opinions of law enforcement officers on cases that have not been solved. The internal validation suggests that the model is prone to overestimating the probability when uncertain, i.e. an estimated probability around 0.4 – 0.6. The initial external evaluation had an Area Under ROC (AUC) score of 0.915. The evaluation against the opinions of law enforcement officers had a much lower AUC, but for cases that had not been solved. The estimations, however, suggests that using the model to filter cases where no link is probable could greatly reduce the time required to compare cases for law enforcement officers. Under favourable conditions, the investigated approach could reduce hours of manual labor to seconds.
9.1 Introduction

Previous research suggests that a large proportion of crimes are committed by a minority of offenders, e.g. in the USA research suggests that 5% of offenders are involved in 30% of the convictions [42]. This claim is supported by Swedish law enforcement agencies. Approximately 22,000 residential burglaries were reported in Sweden during 2012. As some offenders commit multiple crimes, the law enforcement agencies want to detect linked crimes. The detection of linked crimes is helpful to law enforcement for several reasons. Firstly, the aggregation of information from crime scenes allows for an increase in available evidence. Secondly, the investigation of multiple crimes enables a more efficient use of law enforcement resources [37].

Currently, law enforcement officers manually compare residential burglary reports to detect links. The comparison is conducted by looking at the similarity between crimes with regard to modus operandi (MO), spatial proximity, and physical evidence. This manual work process requires law enforcement to allocate time and personnel.

Regression learners have been suggested as a suitable approach to the problem of determining whether the same perpetrator has committed multiple crimes [41]. The work in this paper is based on an approach suggested in previous research [37, 41]. A logistic regression model is a probabilistic regression model, which can be used to predict a binary classification. As such, logistic regression analysis can be used to determine whether two cases are related based on a set of continuous or categorical input variables, which in this case would be burglary similarities. Given a set of training instances with known labels, a model that predicts the binary outcome is constructed. Using this model the potential linkage between crime pairs can be estimated.

A decision support system based on logistic regression to estimate

\(^1\)http://www.bra.se/bra/bra-in-english/home/crime-and-statistics/burglary.html
possible links between cases allows law enforcement officers to find or discard links between cases more efficiently.

9.1.1 Aims and Scope

The purpose is two-fold: first, to investigate whether characteristics differ between linked and unlinked cases. Second, to investigate with which accuracy a logistic regression model is capable of estimating link probability and to investigate the practicality of the predictions. The scope is limited to residential burglaries collected in Sweden during the period of 2012 – 2013.

9.1.2 Contributions

The use of different crime characteristics to differentiate between linked and unlinked crimes is investigated through residential burglary data collected in Sweden during a two-year period. The use of logistic regression analysis is investigated as an approach to estimate link probability between residential burglary pairs and its potential use as a filtering/selection decision support system. As logistic regression analysis uses labeled data to create a model and only a fraction of the studied residential burglaries are labeled, the estimated links of the models are also evaluated for crimes that are unlabeled.

9.2 Background

Since 2011, researchers at Blekinge Institute of Technology have been working with the Swedish Law Enforcement to develop Information and Communication Technology (ICT) targeting residential burglaries. This system is henceforth denoted SAMS. Swedish Law Enforcement is
divided into 21 independent police departments. The National Police Board (NPB) is the central administrative and supervisory authority of the police service. The NPB comprises The National Bureau of Investigation and the Swedish Security Service. In addition, the Swedish police include the National Laboratory of Forensic Science. The 21 police departments operate independently, with one consequence being that limited amount of data is shared between departments.

The SAMS system developed at Blekinge Institute of Technology provides a procedure for dealing with residential burglaries over departmental boundaries. A procedure for structured collection of crime data and a Decision Support System (DSS) have been developed for managing the collected crime data [2]. The structured collection of crime data is done through a digital form. The form requires law enforcement officers to collect certain crime data, e.g. physical position, time, MO characteristics, etc. The usage of a form provides systematic collection of crime data across police departments. A systematic data collection also enables a more robust comparison of residential burglaries in order to detect connections between residential burglaries.

The DSS consists of a web-based interface allowing law enforcement to filter, search, group, and compare residential burglaries according to certain characteristics. An analytical framework is also available, enabling the development of analytical plugins for further analysis.

9.3 Related Work

The problem of linking reported crimes have been investigated previously. Most research, however, have taken place within the fields of psychology or criminology. Crime types of serial characteristics, often with violent aspects, have been the focus of research into linking cases, e.g. sexual offences, rapes, homicides, and different types of burglaries, including violent burglaries [15,37,38,39,40,41].
The research conducted suggests that behavioral consistency is present among offenders and that different offenders behave differently. An individual perpetrator tend to behave with the same modus operandi (MO) in similar situations, i.e. showing behavioral consistency. Different offenders in similar situations tend to behave differently, i.e. showing signs of inter-individual variation [37]. A smaller temporal proximity between situations usually results in an increased similarity for a perpetrator.

Given structured data representing behavioral traits, researchers have computed the similarity between pairs of crimes. Many of the studies have used similarity coefficients between cases, such as the Jaccard coefficient, to represent behavioral consistency [37]. Different aspects of behaviors can be used for comparison, e.g. MO, spatial proximity, and temporal proximity. The MO can be further divided into three domains; entry behavior, target characteristics, and goods stolen [43]. The results have suggested that spatial proximity, and temporal proximity, are better indicators to determine linked crime than the MO characteristics [41]. The similarity scores have been used as input for regression analysis [41,43].

The research that has focused on regression analysis for predicting linked crimes has primarily been investigated using data gathered from smaller geographical areas, e.g. a city. As such, it is interesting to reproduce earlier research, as it has been conducted on a sample from a small geographical area in, e.g. the UK, and “the utility of other predictors may vary across different geo-graphical areas and different samples” [15,41]. As such, it might be that the characteristics considered important are different in two geographical settings, e.g. temporal proximity being more important than spatial proximity. More interestingly, the data in earlier studies are extracted from unstructured crime reports. This aspect is important as a systematic data collection by law enforcement officers guarantees a basic level of information available, whereas extracted data might be incomplete or contain bias. The evaluations of earlier studies have, in most cases, focused on internal validation of the
constructed model. While a few studies have evaluated the model using what can be described as a split sample approach, the evaluation is based on cases that law enforcement officers are able to solve.

9.4 Data

9.4.1 Data Collection

The data set consists of 180 residential burglary reports, collected by law enforcement officers according to the digital form included within the DSS. The form consists of 11 sections and 111 checkboxes and has been developed in cooperation with law enforcement officers. The checkboxes describe, among other aspects, behavioral traits in a structured manner. In addition to the checkboxes, information about time, date and geographical position (longitude, latitude and street address) of the reported incident is also gathered.

The residential burglary reports have been gathered from the southern parts of Sweden and the Stockholm area. Further, law enforcement officers have provided anonymized information about suspects for the residential burglaries, allowing connections between cases to be established. The data contains cases that are considered part of series of burglaries and cases that are considered independent.

9.4.2 Data Preparation

The data is divided into pairs of residential burglary reports. 160 pairs of residential burglary reports are randomly sampled from the dataset, in such a way that 80 of the pairs are linked and 80 are not linked.

The similarity of the pairs are computed for 8 characteristics: All MO behaviors, entry behaviors, property stolen, target (residential) char-
9.5. Method and Experiment Setup

The difference in characteristics between pairs considered linked and unlinked is investigated using Wilcoxon’s signed rank test [115]. Wilcoxon’s signed rank test is used to statistically evaluate whether a statistically significant difference exists between two samples.

Based on the 160 sampled pairs, a logistic regression model is constructed. Logistic regression analysis is a non-parametric procedure that, given one or more independent variables, derives an equation providing a probability that an observation will be a member of a specific group, i.e. the dependent variable. Most commonly, the dependent variable is binary, i.e. comprised of two categories. Logistic regression analysis has been used previously for investigating links between crimes [41,43]. While logistic regression calculates the odds that an observation is a member of a certain category, the category membership is often reported as either 0 or 1 [127].

9.5.1 Evaluation Metrics

The performance of the model is measured using the $R^2$, Somers’ $D_{xy}$ rank correlation, and the C-index ($c$) [127]. $R^2$ indicates how well the instances fit the curve of the regression model. An $R^2$ score of one indicates that the instance can be perfectly mapped to a regression line, whereas a score of zero indicates no such relationship. The C-index is, for binary output models, identical to the Area Under Curve (AUC) [128]. A C-index of 0.5 indicates random classification and a C-index of 1 a per-
fect classification. Somers’ $D_{xy}$ is the rank correlation between two data series, $X$ and $Y$. As such, the rank correlation between a numeric variable (e.g. predictions) and binary variable (e.g. the actual class) can be computed. $D_{xy}$ can be calculated as $D_{xy} = 2(c - 0.5)$. A $D_{xy}$ score of 0 indicates random predictions and a $D_{xy}$ score of 1 indicates perfect predictions.

9.5.2 Evaluation

To validate the model, external and internal validation are used. The internal validation is two-fold. First, bootstrapping based resample validation of the model is used to assess optimism, or overestimation, of the model. Bootstrapping is the preferred technique for evaluating internal model validity, over split sampling and cross-validation [128]. A bootstrap sample is sampled with replacement from the original sample and of the same size as the original sample. Using the bootstrap sample, the model is constructed and then validated using the original sample. The difference indicates the optimism in the model performance. This is repeated 1000 times to obtain stable results. Second, a calibration curve is constructed using bootstrap resampling to investigate possible over- or underestimation of the model. Calibration is the agreement between predicted probability and observed probability. The calibration uses bootstrapping to get estimates for the predicted and observed probability that are corrected for overestimation [128].

External validation is two-fold. First, a new stratified sample with replacement from the original data set is created and the model is evaluated against it. Bootstrap-based validation is preferred before training/test split or cross-validation as, for binary logistic regression models, the latter two put requirements on the sample size that the original data set does not fulfill, e.g. 100 instances belonging to the least frequent category [127]. Secondly, external validation of the model is evaluated using the expertise of law enforcement officers specializing in residential burglaries. Law enforcement officers are asked to evaluate 50 crime
pairs where no information of potential linkage is known. 200 instances are randomly sampled from each region; pairs are constructed and 25 crime pairs are randomly selected from each region. The law enforcement officers’ predictions are then compared to the model’s prediction. The paired cases have two constraints. First, the pairs must be from the same police county. This is because law enforcement officers have limited access to cases outside their county. Second, the cases must be within a three-month period of each other. This due to a unlikelyhood of connections being detected between cases over a longer time period, due to the introduction of unknown variables [126].

9.6 Residential Burglary Characteristics

The distribution of the crime pairs can, for different groups of variables, be seen in Figure 9.1 and Figure 9.2. There is a clear distinction between the linked and unlinked cases with regard to spatial and, to some extent, temporal distance. This can be seen in Figure 9.1a and Figure 9.1b. Similarly, the combined MO distance have a lower distance for the linked than the unlinked (see Figure 9.2a). This separation is something that is hinted at with e.g. entry and victim characteristics as well, and to some small extent the stolen goods characteristics. This is not unexpected, as similar results have been reported using residential burglar reports from the United Kingdom [37].

To further evaluate whether a statistically significant difference exists between the two groups, Wilcoxon’s signed rank test is used. Linked and unlinked cases differ significantly with regards to temporal distance, \( W = 5266.5, p < 0.05 \). There were also a significant difference between linked and unlinked cases for spatial distance \( (W = 5126, p < 0.05) \), combined modus operandi distance \( (W = 4981, p < 0.05) \), entry modus operandi distance \( (W = 4182, p < 0.05) \), stolen goods modus operandi distance \( (W = 4436.5, p < 0.05) \), physical evidence left at the crime scene \( (W = 4013.5, p < 0.05) \), and victim modus operandi distance.
9. Linking Residential Burglaries

Figure 9.1: Distances between linked and unlinked residential burglary pairs with regards to spatial and temporal distances. Linked pairs are blue and unlinked pairs red.

\(W = 4594, p < 0.05\). There was, however, no statistically significant difference between the two groups for the target similarity characteristics, \(W = 3584.5, p > 0.05\). Consequently, the use of the different MO characteristics to differentiate between linked and unlinked crimes is promising. The exception to this seems to be the target MO characteristics.

An example of a pair of residential burglaries that can be considered linked can be observed in Table 9.1. The table contains the different MO characteristics distances, spatial and temporal distances, as well as a visual representation of the spatial distance.

9.7 Results

The initial internal validation results of the model are acceptable, \(R^2 = 0.853, c = 0.980, D_{xy} = 0.959\). Bootstrap based resample validation was
9.7. Results

Figure 9.2: Distances between linked and unlinked residential burglary pairs with regard to MO characteristics. Linked pairs are blue and unlinked pairs red.
9. **Linking Residential Burglaries**

![Graphs showing distances between linked and unlinked residential burglary pairs with regard to MO characteristics.](image)

**(e) Victim modus operandi distance**   
**(f) Evidence distance**

Figure 9.2: Distances between linked and unlinked residential burglary pairs with regard to MO characteristics. Linked pairs are blue and unlinked pairs red.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal</td>
<td>79 days</td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td>11.383 kilometers</td>
<td></td>
</tr>
<tr>
<td>Combined</td>
<td>0.380</td>
<td></td>
</tr>
<tr>
<td>Entry</td>
<td>0.483</td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>0.424</td>
<td></td>
</tr>
<tr>
<td>Goods</td>
<td>0.471</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td>0.182</td>
<td></td>
</tr>
<tr>
<td>Victim</td>
<td>0.353</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.1: Distance between characteristics for a randomly chosen linked pair of residential burglaries.
used to assess the optimism, or overestimation, of the model. The bootstrap evaluation found that the optimism for $D_{xy}$ and $R^2$ of the model was 0.0180 and 0.0395. Based on this, the corrected scores for the model is calculated, $R^2 = 0.814, c = 0.975, D_{xy} = 0.941$. The corrected C-index is calculated as $(1 + D_{xy})/2$ based on the corrected $D_{xy}$.

Evaluation of the model was also done by computing the calibration curve. The calibration results can be seen in Figure 9.3. The model overestimates the predicted probability between $0.4 - 0.6$, but otherwise follows the ideal curve. There are no underestimations visible. Overestimates are preferable in the model, especially around 0.5, as it produces false positives. Underestimates at the same area would produce false negatives which, when dealing with pairing residential burglaries, are to be avoided. The distribution of the predicted probability can also be observed at the top of Figure 9.3, showing that the model predicts mostly high or low probabilities.

The external validation is two-fold. First, a data set was created using stratified sampling with replacement (N=40), henceforth denoted NS. The constructed model was used to predict the linkage status between pairs in the new sample NS. The model was correctly able classify 90% of the instances, see Table 9.2. The model, for the NS set has an AUC score of 0.915. However, it should be noted that this is a bootstrapped sample and, as such, instances might be present during training of the model. While most of the erroneous predictions are false positives, one false negative classification exists.

The second form of external validation was done using data where no labeled information exists, i.e. cases that have not been solved. Law enforcement officers have, however, provided expert opinions for the data, and as such the model is evaluated against the opinions and experience of law enforcement officers. It should be noted that the expert opinions provided by law enforcement officers denote whether they believe that a connection might exist. The constraints on the data were previously explained in Section 9.5. Two data sets are used in the evalu-
9. **Linking Residential Burglaries**

Figure 9.3: Calibration plot

atation, for two police counties in south Sweden. The results of the model prediction can be observed in Table 9.2. Given the results in Table 9.2, the model predicts probable connections different from law enforcement officers. Observing the predictions plotted against each other in Figure 9.4, the problematic cases are the false negatives. It should, however, be noted that there exists no labeled data and that the possible links were only 66% probable. A link between the cases might exist, but law enforcement officers are unsure.

Additionally, the law enforcement officers were asked to estimate, for each pair, how long it took to evaluate whether a link might exist. The mean time, over 25 pairs, were 8.16 and 5 minutes per pair, for Region
9.7. Results

<table>
<thead>
<tr>
<th>Data set</th>
<th>AUC</th>
<th>$D_{xy}$ (SD)</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>0.915</td>
<td>0.830 (0.111)</td>
<td>19</td>
<td>17</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Region 1</td>
<td>0.441</td>
<td>−0.118 (0.173)</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Region 2</td>
<td>0.239</td>
<td>−0.522 (0.175)</td>
<td>0</td>
<td>11</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 9.2: Prediction scores of the model on related data.

1 and Region 2 respectively. This shows that to evaluate 25 case pairs, law enforcement officers will spend between 2 and 3.4 hours. Given that, in Region 1 alone, at least 1200 residential burglaries occurred in 2013, the resources required to review links between cases might not always exist.

Figure 9.4: The predicted probability that a pair of burglaries are linked for law enforcement versus model for two counties in Sweden. Each point represents a case pair. Higher value on x and y-axis represents a higher probability of link.
9. Linking Residential Burglaries

9.8 Discussion

The model has when it comes to internal validation more than acceptable results, even when adjusted for overestimation. The internal validation AUC score of the model is 0.980, indicating suitable predictions. This, however, should only be seen as an upper limit of the model’s capabilities. When encountering new data, lower scores are expected as human behaviour change over time.

The internal evaluation suggests that the model is prone to overestimating the probability of cases when the model is uncertain (when the linkage probability is within the range $0.4 - 0.6$). Overestimation is acceptable within these ranges as it means that the model is suggesting links between cases when it is uncertain. If used as a selection-/filtering tool, suggesting links between cases that are not connected is better than suggesting that there are no links, as cases is not erroneously discarded.

The model is intended to be used as a support tool for an initial filtering of cases that might be linked. As such, merely accepting that the pairs suggested as linked are actually linked is unadvisable. As previously discussed, the model when uncertain tends to overestimate the probability of a connection. The overestimation of the model and the fact that the target domain deals with people and law means that it is strongly advisable that law enforcement officers evaluate the links suggested by the model.

When investigating any crime linkage based on labeled data, the data used to train and test any research approach is based on solved cases. As such, the labeled cases might skew towards certain type of offenders or have other biased characteristics, e.g. local offenders that are previously known. This can explain the results of the second external evaluation, where the model and the law enforcement officers differed in opinions with regard to which pairs were connected. The law enforcement officers were, for positive responses, in the best case unsure about their prediction. For the negative predictions, the predictions of the model
and the law enforcement officers had a higher degree of similarity. The results, however, suggests that using the model for filtering away cases that are not connected could greatly reduce the required time for comparing case reports. As such, resources could be allocated to intensified analysis of the potentially linked cases, or other duties.

If the information concerning the category of a criminal (e.g. juvenile offenders, one-time offenders, etc.), models could be trained for these different categories. Models for different criminal categories could potentially provide more accurate predictions of probable links. However, this data is not currently available.

Similarly, the sample size is quite small, $N < 200$, and generalizability of the model would be increased by having a larger sample size, to increase the time windows and geographical range. An increase in the amount of available data could potentially allow for the creation of county specific models as well, which might be better suited to the characteristics of the specific county.

Given that data becomes increasingly available, the model can be updated. This would allow the model to adjust for changes in MO over time. There would be, given an extensive amount of updates, a risk that the model has been overfitted. Consequently, it would be important to rebuild the model occasionally with a new sampled data set or test the model for overfitting.

9.9 Conclusion

Link estimation using logistic regression analysis was investigated for residential burglaries. Given a residential burglary, the probability that other cases would be linked can be estimated. A model was constructed using data gathered from three different police counties in Sweden during 2011 – 2013. The constructed model estimated the probability be-
9. Linking Residential Burglaries

between pairs of residential burglaries with an AUC of 0.915. These results suggest the feasibility of the approach to help law enforcement officers filter/prioritize residential burglaries for further analysis. In the best case scenario, the time necessary to investigate the link status of a residential burglary is greatly reduced.

There are multiple avenues for extending the work conducted in this study. First, it is hinted in the data and from discussions with law enforcement officers that certain types of offenders are more difficult to apprehend. As such, it would be interesting to investigate multinominal logistic regression to differentiate between different classes of offenders. Secondly, it would seem that there are different characteristics of residential offenders depending on their geographical location, i.e. if they operate in rural communities, major cities, or smaller cities. Given this, the investigation of using local models for different areas could possible provide better probability estimations. Third, a multimodel ensemble classifier could incorporate models for different areas and types of offenders and as such give an more robust estimation.
Abstract

To identify series of residential burglaries, detecting linked crimes performed by the same constellations of criminals is necessary. Comparison of crime reports today is difficult as crime reports are written in unstructured text and often lack a common information-basis. This study investigates the use of clustering algorithms to group similar crime reports based on combined crime characteristics for decision support in residential burglary analysis. The characteristics investigated are modus operandi, residential characteristics, stolen goods, spatial similarity, and temporal similarity. Clustering quality is measured using Connectivity and Silhouette index, stability using Jaccard index, and accuracy is measured using Rand index and a Series Rand index. The performance of clustering using combined characteristics were compared with spatial characteristic. The results suggest that the combined characteristics perform better or similar to the spatial characteristic. In terms of practical significance, the presented clustering approach is capable of clustering cases using a broader decision basis.
10. Combining Modus Operandi for Clustering Burglaries

10.1 Introduction

Internationally, studies suggest that a large proportion of crimes are committed by a minority of offenders, e.g. in the USA research suggests that 5% of offenders are involved in 30% of the convictions [42]. This is echoed by Swedish law enforcement agencies. Law enforcement agencies, consequently, are required to detect whether a connection exists between crimes, e.g. whether crimes are linked. The detection of linked crimes is helpful to law enforcement for several reasons. Firstly, the aggregation of information from crime scenes allows for an increase in available evidence. Secondly, the joint investigation of multiple crimes enables a more efficient use of law enforcement resources [37].

Previously, clustering has been investigated as a method to group crimes based on characteristics, often spatial characteristics. Recently other characteristics have been investigated as well, on an individual basis [2]. Research into estimating linkage using regression analysis have suggested that a combination of characteristics provide a higher accuracy in linkage estimation. This study investigates a combined characteristics distance metric for the use in clustering residential burglaries. Clustering residential burglaries based on different similarity aspects would potentially allow clustering solutions with a better accuracy and a broader decision basis than individual characteristics. Similarly, it would potentially allow law enforcement to find series whilst reviewing a smaller amount of residential burglaries, i.e. used as a case selection DSS. Consequently, the use of this distance metric would allow law enforcement agencies to save resources, whilst providing individual investigators with increased support.

10.1.1 Purpose Statement

The purpose of this study to investigate the effectiveness of a combined distance metric compared to a spatial distance metric. Similarly, the
10.2 Related Work

effectiveness of different clustering algorithms are also investigated. The clustering quality is measured using multiple evaluation metrics and evaluated using statistical tests. A modified version of the Rand index is used to better reflect the clustering solutions accuracy with regard to series of residential burglaries. The data comprises residential burglaries from southern Sweden and the Stockholm area.

10.1.2 Outline

Section 10.2 presents the related work. Section 10.3 and Section 10.4 explains the data and the methodology. The results presented in Section 10.5 and analysed in Section 10.6. Finally, the results are discussed in Section 10.7 and the conclusions of the article presented in Section 10.8.

10.2 Related Work

Connecting crime cases have been investigated before, primarily estimating whether pairs of crime cases are connected. The pair estimation has mostly been conducted for violent crimes with a high possibility for series [15, 37, 38, 39, 40, 41]. But research has also been conducted into clustering crime cases as a means of reducing the number of cases law enforcement officers have to analyze when looking for possible series of crimes [2]. The clustering has been investigated for e.g. residential burglaries. Hotspot detection is a commonly used technique to group cases based on spatial information to predict future crime locations [30, 31, 32, 33, 34, 35]. The clustering research and pair wise link estimation, however, investigated using other crime characteristics, beside spatial information.

There exists multiple crime characteristics which can be used for comparison, e.g. modus operandi (MO), spatial proximity, and temporal proximity. The MO can be further divided into three domains; entry
behaviour, target characteristics, and goods stolen [43]. Entry behaviour describes the procedure used to enter the premises. Target characteristics describes characteristics of the residence being targeted.

Studies have computed the similarity between pairs of crimes based on various crime characteristics. Many of these studies have used similarity coefficients between cases, such as the Jaccard coefficient [37]. Previous research have suggested that there is a difference between the similarities of linked and unlinked residential burglaries, when investigating pairs of crimes [38, 41, 42, 43].

While the clustering research earlier have investigated different crime characteristics independently, pair wise link estimation found that there are reasons to combine multiple characteristics [41]. This has been suggested to increase the accuracy of clustering based solutions for grouping residential burglaries.

10.3 Data

The data set consists of 180 residential burglary reports, collected by law enforcement officers according to a two page digital form that consists of 111 checkboxes together with input fields concerning temporal and spatial data, i.e. time, date and geographical position (latitude, longitude and address). We have developed the form in close cooperation with law enforcement officers and it includes both collected knowledge from crime analysts as well as relevant theory in the field. The form is structured into 11 subsections, that capture data about, among other, the MO, residence properties, preventive measures taken by the plaintiff, and traces left at the crime scene. By using the form a structured data collection process is introduced where the use of the form increase the quality of the collected data compared to traditional open text reports. Mainly because the form works as a checklist that guides the law enforcement officers though mandatory questions to ask. Another positive
effect that comes from using the form, is due to the tick-based checkbox layout, which instantly discretized the collected data, making it more easily interpreted by suitable analysis algorithms.

Once a form is filled out, it is automatically verified and the law enforcement officer is notified on any inconsistencies. When the automatic verification process is passed, and any inconsistencies are addressed, the form is registered in a database and made accessible through a custom developed software-based analysis system. In March 2014 there were approximately 6000 residential burglary forms stored in the database, all collected in the southern part of Sweden and the Stockholm area. More details regarding both the form and the associated analysis system is available in Borg et al [2].

In addition to the data collected in the form, law enforcement officers have provided anonymized data about suspects connected to 97 residential burglary forms. Using these labeled burglary forms it is possible to connect cases that share at least one, or more suspects, i.e. linking cases together into series. As shown in Table 10.1 the labeled cases contain repeat offenders accounting for series that include between two and five burglaries that sum up to a total of 75 burglaries. However, the labeled cases also include 22 single offenders that law enforcement officers could only link to a single offence. The reason for including single offenders in the study is because they are used when calculating the Rand evaluation metric further described in Section 10.4.3.

<table>
<thead>
<tr>
<th>Crime series size</th>
<th>Number of series</th>
<th>Proportion of labeled crimes (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>33</td>
</tr>
<tr>
<td>1(^a)</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>97</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

\(^a\) Not actual series but crimes where burglars could only be tied to one single crime.

Table 10.1: Summary of labeled crimes and series.
10.4 Method

This section describes the distance metrics and clustering algorithms that are evaluated using data from the burglary form introduced in the previous section. In the end of this section the evaluation metrics used in the experiments are described.

10.4.1 Distance Metric

Based on checkbox values within the eleven sections of the burglary form it is possible to calculate pair-wise similarity measures between cases using the Jaccard index. Given two cases $C_1$ and $C_2$ it is possible to calculate the resulting Jaccard index by comparing attributes, i.e. the checkbox values, between the two cases according to Equation 10.1.

Note that since a checkbox represents a binary value the equation for calculating the similarity between binary asymmetric attributes is used instead of the traditional Jaccard index.

$$J(C_1, C_2) = \frac{A_{11}}{A_{10} + A_{01} + A_{11}}$$ (10.1)

In equation 10.1, $A_{11}$ represents attributes that are checked, i.e. given a value of 1, in both case $C_1$ and $C_2$. $A_{10}$ and $A_{01}$ represent attributes that are checked in $C_1$ but not in $C_2$, and vice versa. In this study it is rather the distance between cases that is of interest, and as such the Jaccard distance is used instead. The Jaccard distance is complementary to the Jaccard index and is calculated as according to Equation 10.2.

$$d_J(C_1, C_2) = \frac{A_{10} + A_{01}}{A_{10} + A_{01} + A_{11}}$$ (10.2)

By calculating pair-wise Jaccard distances it is possible to compare
10.4. Method

burglary cases with regard to the variables collected. Similarity analysis of burglaries have to a large extent focused on a single variable as basis for estimating the similarity between cases. However, similarity between cases can also be measured using a combination of multiple variables, e.g. both spatial and MO similarity. Studies that have investigated linking crime pairs suggested that a combination of multiple variables performed better than single alternatives.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Metric</th>
<th>Weight</th>
<th>Min</th>
<th>Max</th>
<th>Median</th>
<th>Mean (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial distance</td>
<td>Kilometers</td>
<td>1.025</td>
<td>0.0</td>
<td>558.140</td>
<td>196.944</td>
<td>248.061 (229.442)</td>
</tr>
<tr>
<td>Temporal distance</td>
<td>Days</td>
<td>1.072</td>
<td>0.0</td>
<td>462.0</td>
<td>150.0</td>
<td>121.215 (95.274)</td>
</tr>
<tr>
<td>Target selection</td>
<td>Jaccard</td>
<td>0.0</td>
<td>0.0</td>
<td>0.682</td>
<td>0.545</td>
<td>0.353 (0.135)</td>
</tr>
<tr>
<td>Entrance method</td>
<td>Jaccard</td>
<td>4.799</td>
<td>0.0</td>
<td>0.737</td>
<td>0.677</td>
<td>0.452 (0.134)</td>
</tr>
<tr>
<td>Stolen goods</td>
<td>Jaccard</td>
<td>2828.618</td>
<td>0.0</td>
<td>0.695</td>
<td>0.631</td>
<td>0.357 (0.151)</td>
</tr>
<tr>
<td>Victim behaviour</td>
<td>Jaccard</td>
<td>15.899</td>
<td>0.0</td>
<td>0.695</td>
<td>0.631</td>
<td>0.357 (0.151)</td>
</tr>
<tr>
<td>Physical trace</td>
<td>Jaccard</td>
<td>2884.257</td>
<td>0.0</td>
<td>0.842</td>
<td>0.642</td>
<td>0.402 (0.181)</td>
</tr>
</tbody>
</table>

Table 10.2: Data characteristics

In this study a multivariate distance metric is investigated as basis for evaluating similarity between cases. The multivariate distance metric is a Weighted Euclidean distance that is calculated from the compounding variables shown in Table 10.2. The weights of each variable is based on the coefficients from a logistic regression analysis model that is developed using the sampled data, as per previous research [41]. The total weighted combined Euclidean distance, \( d_{combined} \), is calculated according to Equation 10.3, where \( D_{spatial} \), \( D_{temporal} \), \( D_{target} \), \( D_{entrance} \), \( D_{goods} \), \( D_{victim} \) and \( D_{trace} \) are the included variables, and \( w_1, w_2, w_3, w_4, w_5, w_6 \) and \( w_7 \) are the associated weights, as presented in Table 10.2.

\[
d_{combined} = \sqrt{w_1(D_{spatial})^2 + w_2(D_{temporal})^2 + \ldots + w_7(D_{trace})^2} \quad (10.3)
\]

Methods that rely on weighted combinations of variable distances offer opportunities within crime analysis, as law enforcement officers could modify these weights to fit the analysis at hand. Although the
weights will be subjective, and affected by the reasoning of the law enforcement officers, it might be advisable to alter the weights for different variables in some analyses. The combined metric, $d_{combined}$, is compared against a spatial distance metric [2].

### 10.4.2 Clustering Algorithms

In this subsection the four clustering algorithms used to evaluate the premise are presented. The algorithms are chosen either because they are widely used or because related studies have indicated the suitability. Whilst the K-means clustering algorithm is one of the more popular algorithms, it does not function reliably on binary data. Consequently, the K-means clustering algorithm was excluded. The default implementation of the four clustering algorithms within the Weka machine-learning software suite\(^1\) were used, except for the Cut-clustering algorithm since it was not included in Weka. The Cut-clustering algorithm was therefore implemented according to the specification [101].

The Cut-clustering algorithm is a graph-based clustering algorithm that rely on minimum cut tree algorithms to cluster the input data, which is represented by an undirected similarity graph [101]. Each node in the graph is an instance and these nodes are connected if the similarity between the corresponding instances is positive, and if so the edge is weighted by the corresponding similarity score. The algorithm works by adding the artificial node $\alpha$ to the existing graph and then connecting all nodes in the graph with it. Then a minimum cut tree is computed and the artificial node removed. The clusters consist of the nodes connected after the artificial node has been removed. A high $\alpha$ value, results in a higher number of clusters produced, and vice versa. Using a binary search approach it is possible to find the $\alpha$ value producing a specific number of clusters.

\(^1\)http://www.luigidragone.com/software/spectral-clusterer-for-weka/
The Expectation-Maximisation (EM) clustering algorithm is a probability-based clustering algorithm [11, 107]. A set of $k$ probability distributions assign attributes to instances within the a priori decided $k$ clusters. The clustering process is two-fold, first the initial values of the means and standard deviations for each of the $k$ probability distributions are guessed. Then, each probability that an instance belongs to each cluster is calculated. Secondly, the means and standard deviation of each cluster distribution is recalculated based on the latest clustering result. This process is continued until the classes that instances are assigned to remains unchanged, which means the EM clustering algorithm have converged to a maximum. Unfortunately, this might be a local instead of the global maximum. Therefore, the whole process is repeated multiple times, with different initial estimate values of the means and standard deviations, to increase the chance of finding the global maxima. Finally, the largest maxima is selected and its related $k$ probability distributions are used in any further clustering.

Hierarchical clustering algorithm is implemented using a either a top-down or bottom-up (agglomerative) approach [11]. The agglomerative approach, begins by considering each instance as its own cluster. Next, the two clusters with the least distance between them are identified and merged together into one new cluster. Then, the process of finding the two closest clusters and merging them is continued until only one final cluster exists. The output of the clustering is the sequence of mergings that could be represented as a hierarchical clustering structure in the form of a binary tree (dendrogram). A key part of the Hierarchical clustering algorithm concerns the distance calculation between clusters. Several different methods are available, such as the single-linkage method that makes use of the minimum distance between two clusters, which also makes it sensitive to outliers. Another method is the centroid-linkage that calculates the centroid of a cluster based on its members’ internal distances, and then use the distance between centroids to determine the closest clusters. The complete-linkage method computes the maximum distance between two clusters [11]. The adjusted complete-linkage method, similar to the complete linkage-method, computes the
maximum distance between two nodes from two clusters. The method then finds the largest distance between nodes within either of the two clusters and subtracts that from the maximum distance between the two clusters [11]. The Hierarchical clustering algorithm in this paper uses three different approaches to calculate the distance between clusters, single-link, complete-link, and adjusted complete-link.

Spectral clustering is a graph-based clustering algorithm that has been found to generally detect good clustering solutions [12, 106]. The algorithm takes number of clusters and a similarity matrix as input, and calculates an $n \times n$ affinity matrix for $n$ instances [106]. Using Principle Component Analysis it is possible to identify relevant Eigenvalues and their associated Eigenvectors. Next, the Eigenvectors with sufficiently large Eigenvalues are extracted, and the number of extracted Eigenvectors are equal to the number of dimensions in the data set. Finally, dimension reduction is carried out by mapping the extracted Eigenvectors into a new space where the instances could be more efficiently clustered.

### 10.4.3 Evaluation Metrics

One of the most important aspects of cluster analysis is the validation of clustering results. Research into clustering have indicated that it is not reliable to use a single cluster validation measure. Instead it is preferable to use at least two measures that reflect different aspects of a partitioning. Similar, in this study five different validation measures are implemented. The quality of the clustering solution is estimated using two validity indices, Connectivity and Silhouette index. The connectivity is used for measuring connectedness [111]. The Silhouette Index is used for assessing compactness and separation properties of a partitioning [112]. For evaluating the stability of a clustering method, the Jaccard index is used [113]. Rand index and Series Rand index is used for assessing accuracy [114]. This measure is applied to calculate the agreement between the clustering solution and the known clustering solution. The traditional Rand index is calculated using all instances and the Series
10.4. Method

Rand index is calculated using only the instances that belong in a series.

Connectivity captures the degree to which cases are connected within a cluster by keeping track of whether the neighboring cases are put into the same cluster [111]. Let \( m_{i(j)} \) be the \( j \)th nearest neighbour of case \( i \), and let \( \chi_{im_{i(j)}} \) be zero if \( i \) and \( j \) are in the same cluster and \( 1/j \) otherwise. Then for a particular clustering solution (partition) \( P = \{C_1, C_2, \ldots, C_k\} \) of data set \( M \), which contains \( m \) instances (rows) in \( n \) different experimental conditions (columns), the Connectivity is defined according to Equation 10.4. It has a value between zero and infinity that should be minimized.

\[
Conn(P) = \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{im_{i(j)}} \quad (10.4)
\]

Silhouette index (SI) reflects the compactness and separation of clusters [112]. Let \( P = \{C_1, C_2, \ldots, C_k\} \) be a clustering solution (partition) of data set \( M \), which contains \( m \) cases. Then the Silhouette index is defined according to Equation 10.5. In the equation \( a_i \) represents the average distance of case \( i \) to the other cases of the cluster to which the case is assigned, and \( b_i \) represents the minimum of the average distances of case \( i \) to cases of the other clusters. The Silhouette index vary between -1 to 1 and higher value indicates better clustering results.

\[
s(P) = \frac{1}{m} \sum_{i=1}^{m} (b_i - a_i) / \max\{a_i, b_i\} \quad (10.5)
\]

The Jaccard index is used to evaluate the stability of a clustering method [113]. The considered clustering method is randomized so it produces different clustering results \( p \) times. The averaged Jaccard index is computed over all \( p(p-1)/2 \) pairs of \( p \) outcomes. The Jaccard index is calculated as follows. Given a pair of clustering solutions of the same data set \( (M) \), \( P_1 \) and \( P_2 \), \( a \) is defined as the number of pairs that belong to the same cluster in \( P_1 \) as well as in \( P_2 \). Let \( b \) be the number
of pairs that belong to the same cluster in $P_1$ but not in $P_2$. Further, $c$ is
defined to be the number of pairs that belong to the same cluster in $P_2$
but not in $P_1$. The Jaccard index between $P_1$ and $P_2$ is then defined as in
Equation 10.6.

$$J(P_1, P_2) = \frac{a}{a+b+c}$$

(10.6)

The Rand index (RI) is used to calculate the accuracy of cluster solu-
tions (partitions). This allows for a measure of agreement between two
partitions, $P_1$ and $P_2$, of the same data set ($M$). Each partition is viewed
as a collection of $m(m-1)/2$ pairwise decisions, where $m$ is the num-
ber of cases. For each pair of cases $g_i$ and $g_j$ in $M$, the partition either
assigns them to the same cluster or to different clusters. Let $a$ be the
number of decisions where $g_i$ is in the same cluster as $g_j$ in $P_1$ and in
$P_2$. Let $b$ be the number of decisions where the two cases are placed in
different clusters in both partitions. Total agreement, thus accuracy, can
then be calculated using Equation 10.7. The Rand index ranges between
0 to 1, where a higher value indicates a higher accuracy. $P_2$ is known
beforehand and based on labeled data.

$$Rand(P_1, P_2) = \frac{a+b}{m(m-1)/2}$$

(10.7)

The Series Rand index (SRI) is used to calculate the accuracy, but with
emphasis on series. This is implemented similar to the traditional Rand
index, but instead only measures the agreement of two clustering solu-
tions with regard to cases that are part of a series, i.e. disregarding from
crimes that doesn’t belong to a series.

## 10.5 Results

The results are presented in four $m \times n$ matrixes (one for each metric) per
algorithm and distance measure. The Cut-clustering algorithm failed to
produce non-trivial clustering solutions when using the combined distance metric, and only produced non-trivial clustering solutions in 50% percent of the runs when using the spatial distance metric. As such, there are no metrics available for the Cut-clustering algorithm when using the combined distance metric. The Connectivity (Table 10.4) and Silhouette index (Table 10.3) indicate the clustering quality. The measured Silhouette index can be seen in Table 10.3. It seems that while the Spectral clustering algorithm performs better using the combined metric, the Silhouette index of the other algorithms are quite similar.

<table>
<thead>
<tr>
<th></th>
<th>Combined(_1)</th>
<th>Combined(_2)</th>
<th>Spatial(_1)</th>
<th>Spatial(_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>—</td>
<td>—</td>
<td>-0.46</td>
<td>-0.18</td>
</tr>
<tr>
<td>EM</td>
<td>0.81</td>
<td>0.86</td>
<td>0.82</td>
<td>0.88</td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td>0.46</td>
<td>0.44</td>
<td>0.47</td>
<td>0.46</td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>0.46</td>
<td>0.45</td>
<td>0.48</td>
<td>0.44</td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td>0.46</td>
<td>0.45</td>
<td>0.48</td>
<td>0.47</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.66</td>
<td>0.62</td>
<td>0.50</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table 10.3: Mean Silhouette index for the algorithms and distance functions

<table>
<thead>
<tr>
<th></th>
<th>Combined(_1)</th>
<th>Combined(_2)</th>
<th>Spatial(_1)</th>
<th>Spatial(_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>—</td>
<td>—</td>
<td>49.50</td>
<td>99.00</td>
</tr>
<tr>
<td>EM</td>
<td>90.70</td>
<td>97.50</td>
<td>90.70</td>
<td>97.50</td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td>85.80</td>
<td>91.20</td>
<td>92.50</td>
<td>86.30</td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>96.70</td>
<td>97.40</td>
<td>97.80</td>
<td>96.50</td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td>87.10</td>
<td>94.20</td>
<td>82.60</td>
<td>95.60</td>
</tr>
<tr>
<td>Spectral</td>
<td>98.20</td>
<td>97.90</td>
<td>97.40</td>
<td>96.80</td>
</tr>
</tbody>
</table>

Table 10.4: Mean Connectivity index for the algorithms and distance functions

The Connectivity index doesn’t show any distinct differences between the spatial and combined metric. In fact, for the Hierarchical clustering algorithm there is only minor difference between the two distance functions, as can be observed in Table 10.4. Table 10.5 and Table 10.6 show the accuracy of the clustering solutions measured by the Rand index and Series Rand index respectively. For both metrics there
10. Combining Modus Operandi for Clustering Burglaries

<table>
<thead>
<tr>
<th></th>
<th>Combined₁</th>
<th>Combined₂</th>
<th>Spatial₁</th>
<th>Spatial₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>—</td>
<td>—</td>
<td>0.04</td>
<td>0.09</td>
</tr>
<tr>
<td>EM</td>
<td>0.96</td>
<td>0.97</td>
<td>0.96</td>
<td>0.97</td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td>0.89</td>
<td>0.92</td>
<td>0.89</td>
<td>0.92</td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td>0.91</td>
<td>0.95</td>
<td>0.91</td>
<td>0.95</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 10.5: Mean Rand index for the algorithms and distance functions

<table>
<thead>
<tr>
<th></th>
<th>Combined₁</th>
<th>Combined₂</th>
<th>Spatial₁</th>
<th>Spatial₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>—</td>
<td>—</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>EM</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td>0.85</td>
<td>0.87</td>
<td>0.85</td>
<td>0.87</td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td>0.86</td>
<td>0.92</td>
<td>0.86</td>
<td>0.92</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.93</td>
<td>0.94</td>
<td>0.94</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 10.6: Mean Series Rand index for the algorithms and distance functions

<table>
<thead>
<tr>
<th></th>
<th>Combined₁</th>
<th>Combined₂</th>
<th>Spatial₁</th>
<th>Spatial₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>—</td>
<td>—</td>
<td>0.47</td>
<td>0.65</td>
</tr>
<tr>
<td>EM</td>
<td>0.45</td>
<td>0.59</td>
<td>0.45</td>
<td>0.59</td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td>0.10</td>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>0.21</td>
<td>0.19</td>
<td>0.21</td>
<td>0.19</td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td>0.10</td>
<td>0.12</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Spectral</td>
<td>0.31</td>
<td>0.30</td>
<td>0.22</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 10.7: Mean Jaccard index for the algorithms and distance functions

are only negligible differences between the combined and spatial metric, but the Series Rand index shows a lower score than the Rand index. This is because the accuracy of the clustering solutions are not inflated by crimes not part of a series, as the Series Rand index only includes crimes part of a series of residential burglaries. Table 10.7 shows the stability of the clustering algorithms for the different data sets using the Jaccard index. The Jaccard index is used to indicate the stability of the clustering solutions. The EM algorithm shows best performance with a
Jaccard index of around 0.5. The Cut-clustering algorithm only produced non-trivial clustering solutions using the combined metric, and it produced trivial clustering solutions in 50% of the cases when using the spatial metric. Therefore the results of Cut-clustering algorithm for the Jaccard metric should be discarded. The Spectral algorithm produce better using the combined metric compared to the spatial, around 0.3 and 0.2 respectively.

10.6 Result Analysis

The results evaluation is two-fold. First, the difference between the algorithms performance for the two distance functions are evaluated using Wilcoxon’s test. Second, the performance of the different algorithms are evaluated using Friedman’s test. The algorithm that has the best mean performance over multiple evaluation metrics is investigated further using a Nemenyi post-hoc test.

10.6.1 Distance Metric Comparison

For the Spectral clustering algorithm the combined distance metric was significantly better than the spatial distance metric with regard to Silhouette index \( W = 12, p < 0.05 \), Rand index \( W = 80, p < 0.05 \), Jaccard index \( W = 105, p < 0.05 \), but not for Connectivity \( W = 138.5, p > 0.05 \). With regards to Series Rand index \( W = 400, p < 0.05 \), the spatial distance metric performed significantly better. This can be observed in Figure 10.1, Figure 10.2, Figure 10.4, and Figure 10.3 where the observations of the Spectral clustering algorithm for both data samples has been visualized using box-plots. While there are some outliers, the figures shows that the two distance functions do not overlap. A significant difference were detected for the Hierarchical Clusterer (Single) clustering algorithm \( W = 278, p < 0.05 \) with regard to the Silhouette index, but
not for the other metrics. There were found to be no significant difference between the distance functions for Hierarchical Clusterer (Single) or Hierarchical Clusterer (Complete) clustering algorithms. Since EM doesn’t use a distance metric, there were no reason to test this. As the Cut-clustering algorithm failed to produce clustering solutions for the combined distance metric, it must be concluded that the spatial distance metric is preferable in that case.

![Figure 10.1: Silhouette index per distance metric for the Spectral clustering algorithm, indicating cluster solution quality.](image1)

![Figure 10.2: Rand index per distance metric for the Spectral clustering algorithm, indicating cluster solution accuracy.](image2)
10.6. Result Analysis

Figure 10.3: Series Rand index per distance metric for the Spectral clustering algorithm, indicating cluster solution accuracy.

Figure 10.4: Jaccard index per distance metric for the Spectral clustering algorithm, indicating cluster solution stability.

10.6.2 Algorithm Comparison

Friedman’s test were applied to the different metrics to evaluate whether any algorithm performed significantly better than another algorithm. Friedmans test found significant differences between the algorithms for the Rand index ($\chi^2 = 14.428, df = 3, p < 0.05$) and the Series Rand index ($\chi^2 = 12.149, df = 3, p < 0.05$). The test found no significant differences for the Silhouette index ($\chi^2 = 1.75, df = 3, p > 0.05$) or the Connectivity index($\chi^2 = 12.28, df = 3, p > 0.05$). Friedman’s test found no significant difference for the Jaccard index ($\chi^2 = 6.473, df = 3, p > 0.05$).
The Nemenyi test for the Rand index shows that in this case, the Spectral clustering algorithm performed significantly better than the Cut clustering algorithm and the Hierarchical Clustering algorithm (using an adjusted complete link approach) at $p = 0.05$ and $p = 0.01$ respectively (Table 10.8). The Hierarchical clustering algorithm (using a complete link approach) also performed significantly better than the Cut clustering algorithm. For the Series Rand index, Friedman’s test found that the Spectral clustering algorithm performed significantly better than the Cut clustering algorithm and the Hierarchical Clustering algorithm (using an adjusted complete link approach) at $p = 0.05$ and $p = 0.01$ respectively (Table 10.9). No significant difference can be detected between the other algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Cut</th>
<th>EM</th>
<th>$HC_1$</th>
<th>$HC_2$</th>
<th>$HC_3$</th>
<th>Spectral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>**</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average Rank 6 3 5 2 4 1

Critical difference at $p = 0.05: 3.769$, Critical difference at $p = 0.01: 4.449$

* denotes significant difference at $p = 0.05$, ** denotes significant difference at $p = 0.01$

$HC_{1-3}$: HierarchicalClusterer (Adj. Complete), HierarchicalClusterer (Complete), and HierarchicalClusterer (Single)

Table 10.8: Nemenyi test results for Rand index

### 10.6.3 Evaluation Metric Analysis

A correlation matrix between the variables was investigated to see if there were any unlabeled evaluation metrics that could be used to indicate a higher Rand index. Table 10.10 and Table 10.11 shows how the different variables correlate to each other for the spatial and combined distance functions. Similar to the box-plots (Figure 10.1-10.3), the data is
Table 10.9: Nemenyi test results for Series Rand index

<table>
<thead>
<tr>
<th>Cut</th>
<th>EM</th>
<th>HC₁</th>
<th>HC₂</th>
<th>HC₃</th>
<th>Spectral</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Adj. Complete)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Complete)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HierarchicalClusterer (Single)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>**</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average Rank: 6 2.5 5 2.5 4 1

Critical difference at $p = 0.05 : 3.769$, Critical difference at $p = 0.01 : 4.449$

* denotes significant difference at $p = 0.05$, ** denotes significant difference at $p = 0.01$

HC₁₋₃: HierarchicalClusterer (Adj. Complete), HierarchicalClusterer (Complete), and HierarchicalClusterer (Single)

limited to the observations for the Spectral clustering algorithm. As can be expected, the Rand index and Series Rand index closely correlate to each other regardless of the distance metric. The Connectivity correlates negatively to the Rand index and Series Rand index, also independent of distance metric. This correlation is not surprising as a lower connectivity indicates a better cluster solution. For the combined distance metric, there is a positive correlation, albeit small, between the Silhouette index and Rand index. Surprisingly, there is a negative correlation between the Series Rand index and Silhouette index. This would indicate that, for the spatial distance metric, a cluster solution which have problems separating clusters simultaneous have a higher accuracy.

There is no clear metric that have a high correlation with either the Rand index or the Series Rand index. As such, using a evaluation metric which relies on unlabeled data to indicate a high accuracy seems to be without basis.
10. Combining Modus Operandi for Clustering Burglaries

<table>
<thead>
<tr>
<th></th>
<th>Connectivity</th>
<th>Silhouette index</th>
<th>Rand index</th>
<th>Series Rand index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity</td>
<td>1.00</td>
<td>0.10</td>
<td>-0.07</td>
<td>-0.07</td>
</tr>
<tr>
<td>Silhouette index</td>
<td>0.10</td>
<td>1.00</td>
<td>0.11</td>
<td>-0.06</td>
</tr>
<tr>
<td>Rand index</td>
<td>-0.07</td>
<td>0.11</td>
<td>1.00</td>
<td>0.97</td>
</tr>
<tr>
<td>Series Rand index</td>
<td>-0.07</td>
<td>-0.06</td>
<td>0.97</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 10.10: Correlation matrix for the Combined distance metric

<table>
<thead>
<tr>
<th></th>
<th>Connectivity</th>
<th>Silhouette index</th>
<th>Rand index</th>
<th>Series Rand index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity</td>
<td>1.00</td>
<td>0.27</td>
<td>-0.13</td>
<td>-0.16</td>
</tr>
<tr>
<td>Silhouette index</td>
<td>0.27</td>
<td>1.00</td>
<td>-0.52</td>
<td>-0.53</td>
</tr>
<tr>
<td>Rand index</td>
<td>-0.13</td>
<td>-0.52</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Series Rand index</td>
<td>-0.16</td>
<td>-0.53</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 10.11: Correlation matrix for the Spatial distance metric

10.7 Discussion

The results and the analysis showed that the combined distance metric performed as good as or in certain cases better than the spatial distance metric. While there were exceptions to this, the difference between the two in those cases were negligible. There are advantages to the combined distance metric that are not available to a single characteristics distance metric.

An advantage is the increased amount of information used. While the spatial distance metric performs with similar results to the combined distance metric, it could be argued that increasing the amount of information the clustering solution is based on allows more robust decision making support. Also, while spatial analysis of residential burglaries or other types of crimes, i.e. hotspot analysis, can be a good indicator of crimes part of a series or indicating crime waves, there is no possibility of identifying series of crimes committed over a longer time period or identifying a series within a high risk area where multiple criminals operate frequently. In these cases other information must be included, e.g. MO information. Whilst this can be done manually by law enforcement
officers, manual analysis is often resource demanding, often limited to e.g. violent crimes.

A second advantage to the combined distance metric is that it would allow law enforcement officers to provide their own weights to the different characteristics based on their expert opinions. Providing clustering solutions that can be deemed to be adapted to each individual investigation. However, default weights can be provided based on solved crimes. A drawback of basing the default weights on solved cases would be that they are biased towards the cases that law enforcement are able to solve. At the moment, that is cases that have a close spatial and temporal distance. This could be remedied using organizational improvement, something that e.g. Swedish Law Enforcement is currently working on.

A potential drawback to the combined distance metric is that not all clustering algorithms can be used with it. This is due to the inclusion of binary data in the instances. Algorithms such as the K-means clustering algorithm requires non-binary data. However, the Spectral clustering algorithm seems to be a good candidate. The Spectral clustering algorithm performed significantly better than the Cut-clustering and Hierarchical clustering algorithm regardless of which distance metric used.

When evaluating clustering solutions with multiple singletons, the True Negatives inflates the Rand index. This is also true of clustering solutions with multiple smaller clusters. The Series Rand index provides accuracy based on how well the series have been clustered, without taking into account crimes not part of any series. The Series Rand index, however, is also susceptible to the problem of multiple small clusters, albeit to a lesser extent than the Rand index.

There is no cluster evaluation metric that have a high correlation with either the Rand index or the Series Rand index. Consequently, it is not possible from the results to identify an evaluation metric which relies on unlabeled data capable of indicating a high accuracy. This is unfortunate as the amount of labeled data for residential burglaries is likely to be sparse. However, it is our opinion that the Silhouette index is still a
10. Combining Modus Operandi for Clustering Burglaries

reasonable evaluation metric when labeled data is missing. The Silhouette index reflects the compactness and separation of clusters [112]. Each series of residential burglaries should have a high intra-series similarity score and a low inter-series similarity score, which is similar to what the Silhouette index evaluates [37].

10.8 Conclusion

A combined distance metric for clustering residential burglaries have been investigated. The performance was evaluated based on multiple evaluation metrics using five clustering algorithms. The combined distance metric was compared against a spatial distance metric. The combined distance metric generally performed similar or better than the spatial distance metric, but in a few cases it performed worse. However, the combined distance metric has the advantage of using a more complete picture of the residential burglary as the basis for the clustering of the burglary.

The performance algorithms were evaluated using Friedmans test and the Nemenyi test. The Spectral clustering algorithm were the highest ranking algorithm and performed significantly better than the Cut-clustering algorithm and Hierarchical clustering algorithm.

As knowledge of perpetrators is not common, it is argued that the Silhouette index is an reasonable metric to use when evaluating cluster solutions of data without any knowledge of the perpetrators. However, no clear correlation could be found between the Silhouette index and the accuracy indices for the combined distance metric.
10.9 Future work

Two venues for future work has been identified. First, a study based on more labeled data would allow the results to be more generalizable. Second, the approach should be investigated for other crime categories, such as vehicle theft or various frauds. Different crime categories have different behavioural characteristics, and whether clustering can be used to group series of crimes has not been investigated using MO characteristics.

Acknowledgement

This work has been carried out within the project “Computer aided support for increased knowledge about serial crimes”, partially funded by the EU Regional Development Fund. The authors would also like to thank all the members of the project.
References


References


References


178


[90] Seongwook Youn and Dennis McLeod. Efficient Spam Email Fil-
tering using Adaptive Ontology. ITNG ’07. Fourth International

[91] Seongwook Youn and Dennis McLeod. Spam Email Classifica-

[92] W Li, S Hershkop, and S Stolfo. Email archive analysis through
graphical visualization. Proceedings of the 2004 ACM workshop on
Visualization and data mining for computer security, 2004.

[93] Meizhen Wang, Zhitang Li, Ling Xiao, and Yunhe Zhang. Research
on behavior statistic based spam filter. Education Technology and
Computer Science, 2009. ETCS ’09. First International Workshop on,

[94] R Nussbaum, A Esfahanian, and Pang-Ning Tan.;. History-based
e-mail prioritization. ASONAM ’09: Social Network Analysis and

[95] M Hall, E Frank, G Holmes, and B Pfahringer. The weka data

[96] Fabrizio Sebastiani. Classification of text, automatic. The Encyclo-

[97] Tom Fawcett. ROC graphs: Notes and practical considerations for data

[98] Ying Xu, Victor Olman, and Dong Xu. Clustering gene expression
data using a graph-theoretic approach: an application of mini-

[99] Viet-Anh Nguyen and Pietro Lió. Measuring similarity between
gene expression profiles: a Bayesian approach. BMC Genomics,

[100] Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri. SparseDTW:
a novel approach to speed up dynamic time warping. In AusDM
’09: Proceedings of the Eighth Australasian Data Mining Conference.


References


The edges in the graph are represented by different similarity coefficients, making the edge weights a measure of similarity between nodes. The different similarity coefficients are explained in this section. The similarity coefficients have been chosen based on results suggested in previous research.

The Jaccard coefficient (otherwise known as the Jaccard index or Tanimoto coefficient) is a measure of similarity between two pairs, A and B, based on the data shared and the data unique to each set, as shown in Equation A.1. A similarity value between 0 and 1 is computed, where a value of 0 indicates that the two sets are identical.

\[
Jaccard = 1 - \frac{|A \cap B|}{|A \cup B|}
\]  

(A.1)

The Jaccard coefficient is used to compute the similarity between incident reports based on the complete binary data available, data rep-
representing stolen goods and data representing the target, i.e. residential characteristics.

Temporal proximity between instances is also used as a similarity measure. The data gathered by law enforcement officers contains information to compute the temporal proximity between residential burglaries. Due to the nature of these crimes, i.e. the crimes are often committed when the residents are away, the accuracy of the reported occurrence time and dates is often low. Consequently, the reported occurrence is often limited to a day of the week, but reporting would preferable describe when crime occurred within a range of hours. The proximity in time between cases is computed as $A_{time} - B_{time}$ if A occurred after B and vice versa.

Similar to temporal proximity, the spatial proximity between instances is used as a similarity measure. Data gathered by law enforcement officers contains the address where the residential burglary took place, to a degree that allows us to find the longitude and latitude. From the coordinates, the proximity between the longitude and latitude are computed and converted to meters. It should be noted that the distance computed is the shortest path between the two cases, i.e. the geodesic distance.

A.2 Edge Removal Criteria

As discussed in Section 8.4, the minimum cut tree algorithm, when given complete graphs or near-complete graphs, can produce trees that are star-shaped, i.e. each node is connected directly to the root node, or unary. Consequently, it is possible that the clustering can be improved by converting complete graphs into incomplete graphs. Two approaches for this conversion are investigated. In the first approach, several threshold values are computed and the graphs are pruned with these values, only keeping edges where the nodes are considered similar to a certain degree. Threshold edge removal for graph transformation can be con-
sidered a global approach, in that a single threshold value is computed and used for all edges in the graph.

The second approach use time and distance based measures, and given the outcome the edge is removed or the weight is changed to indicate lesser similarity. The distance-based edge removal can be considered local, i.e. only a single edge is investigated at a time. Given this, the criteria for removing an edge can be different for each edge.

**Thresholded edge removal**

Thresholded edge removal for graph transformation can be considered a global approach, in that a single threshold value is computed and used for all edges in the graph. Only edges where the nodes are considered similar to a certain degree, e.g. below the threshold value, are kept. Three different threshold values, and their ability to produce quality clusters, are investigated.

The mean value is the sum of values of the similarity indices of a set of pairs of instances divided by the number of instances. Every edge in the graph, whose value is above the threshold value, is removed.

The quartile value is considered the value separating an ordered set into a number of subsets. The median is the 2-quartile value separating a set divided in to two parts. The quartile value is a set of three values that divide a set into four groups. The different quartiles used are denoted as 2nd quartile ($Q_2$), also known as the median, and 3rd quartile ($Q_3$).

The $Q_2$ value\(^1\) is computed as described in Equation A.2.

\[
Q_2(X) = \begin{cases} 
X_{(N+1)/2} & \text{if } N \text{ is odd} \\
\frac{1}{2}(X_{N/2} + X_{1+(N/2)}) & \text{if } N \text{ is even}
\end{cases} \tag{A.2}
\]

If the number of items in the set is odd, the $Q_2$ value is middle value of the set. If the number of items in the set is even, the $Q_2$ value is the mean value of the two items in the middle of the set.

---

\(^1\)http://mathworld.wolfram.com/StatisticalMedian.html, 2013-02-18
**Distance-based edge removal** differs from the threshold edge removal in that it is based on spatial and temporal proximities (independent of the underlying edge representation). An additional difference is that distance-based edge removal can be considered local. That is, only a single edge is investigated at a time. Given this, the criterion for removing an edge can be different for each edge.

*The Mantel cross product adaption* is based on the Mantel index, which is a correlation test between time and distance for pairs of instances [121]. The Mantel index was designed to alleviate some of the problems with previous indices, where cut off points affect result and results can be significant both if the time/space distance is short or long. It is used to detect correlations between two matrices, and as such needs to be adapted to compare between two instances only. The Mantel index cross product is defined as follows:

\[ T = \sum_{i=0}^{N} \sum_{j=0}^{N} (X_{i,j} - \text{Mean}(X)) \times (Y_{i,j} - \text{Mean}(Y)) \]  

(A.3)

The variables can be explained as follows: \( N \) is the number of instances, \( X \) a set of similarities of one index (e.g. space) between two instances, and \( Y \) a set of similarities of another index (e.g. time) between the same instances. The following equations are used as base to remove edges where time and space has exceeded a certain point in the dataset. The time and space proximity between two instances are compared against the mean time or space proximity. If one of the conditions is negative, the weight of the edge is increased by half. If both conditions are negative, the edge is removed.

\[ (X_{i,j} - \text{Mean}(X)) \times -1 \]  

(A.4)

\[ (Y_{i,j} - \text{Mean}(Y)) \times -1 \]  

(A.5)

*Journey Time Distance (JTD)* is a measure used in Geographical Information Systems (GIS), to determine whether time/space distances
between cases are reasonable [35]. The measure used here is a simplified version that assumes a straight travel distance and a fixed speed. The JTD is determined by calculating distance divided by speed equals time, e.g. whether a criminal reasonably can travel between cases. The equation used to investigate this is as follows:

\[
\frac{X_{i,j}}{100000} > (Y_{i,j} \times 24)
\]

(A.6)

The distance in meters \((X_{i,j})\) between two cases divided by 100,000 (100 km/h) gives the time it would take to travel between the two cases. If that is larger than the temporal proximity \((Y_{i,j})\) the cases are reasonably not connected and the edge is removed.

It has been argued that a temporal proximity no greater than 6 months is the longest period that a dataset should span, as on longer time period the movement of people affects the outcome [126]. This constraint on time span is considered by removing the edge, if the temporal proximity is longer than 3 months.
$B$

Standardised Complaint Form
### B. Standardised Complaint Form

**STANDARDISED COMPLAINT ROUTINE**
Residential Burglary

<table>
<thead>
<tr>
<th>Date of Crime</th>
<th>Registration number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Address**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 1. TIME
- Summer (14 - 21/10) - Winter
- Daytime (05-16) - Evening (16-24)
- Nighttime - Unknown
- Weekend (Sat/Sun/Holiday) - Unknown
- Weekday

#### 2. TYPE OF RESIDENTIAL AREA
- Singular
- Multiple neighbors - Condominium
- Adjacency to forest or field

#### 3. TYPE OF RESIDENCY
- House
- Bungalow
- Tenure/Townhouse
- Co-operative apartment
- Apartment
- Multiple floors
- Top floor apartment

#### 4. ALARM
- Triggered
- Untriggered
- Stalled
- No alarm

#### 5. OBJECT DESCRIPTION
- Litterbox empty
- U/S floors
- A/B outdoors
- Break lighting
- Vehicle in driveway
- Lawn mowed/Snow shooed
- Dog/Sign indicating dog
- None of the above

#### 6. PLAINTIFF
- Home during crime
- Away max 2 hours
- Away 2-24 hours
- Away more than 24 hours
- Entrepreneur
- Appears in company index
- Planned absence
- Spontaneous absence
- Household services (R/T/R/O/T)
- Housecall
- Call from unknown number/person
- Documented absence online
- Children home
- Advertised buy/sell
- Vehicles at airport border
- None of the above

#### 7. ACCESS OBJECT
- Secluded area
- Unobstructed view
- Gallery
- Balcony
- Entrance/Front Door
- Cellar
- Upper floor
- Unknown
- Door
- Window
- Break
- Crushed glass
- Cist excited
- Tools from site
- Disassembled

#### 8. IDENTIFIED SEARCH
- Flaunted
- Locked
- None
- Unknown
<table>
<thead>
<tr>
<th>9. GOODS</th>
<th>10. TRACE EVIDENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alcohol/Tobacco</td>
<td>Fingerprints</td>
</tr>
<tr>
<td>Electronics</td>
<td>Skeletal</td>
</tr>
<tr>
<td>Gold/Jewelry/Cash</td>
<td>Gloves</td>
</tr>
<tr>
<td>Clothing</td>
<td>Shoes</td>
</tr>
<tr>
<td>Pharmaceutical</td>
<td>Tools</td>
</tr>
<tr>
<td>Toys</td>
<td>Mainsprings</td>
</tr>
<tr>
<td>Weapons</td>
<td>Goods for trace evidence investigation</td>
</tr>
<tr>
<td>Ball/Alarm box</td>
<td>No trace evidence</td>
</tr>
<tr>
<td>Perfume</td>
<td></td>
</tr>
<tr>
<td>Keys</td>
<td></td>
</tr>
<tr>
<td>Vehicles</td>
<td></td>
</tr>
<tr>
<td>Passport ID</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>Nothing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11. Other:
- Money
- Tips
- Traceable goods
- None of the above

Registration number:

Date: 2012-12-04 14:23:51

RPS 403.24 2011-01-01/1

193
ABSTRACT

Law enforcement agencies regularly collect crime scene information. There exists, however, no detailed, systematic procedure for this. The data collected is affected by the experience or current condition of law enforcement officers. Consequently, the data collected might differ vastly between crime scenes. This is especially problematic when investigating volume crimes.

Law enforcement officers regularly do manual comparison on crimes based on the collected data. This is a time-consuming process; especially as the collected crime scene information might not always be comparable. The structuring of data and introduction of automatic comparison systems could benefit the investigation process. This thesis investigates descriptive and predictive models for automatic comparison of crime scene data with the purpose of aiding law enforcement investigations.

The thesis first investigates predictive and descriptive methods, with a focus on data structuring, comparison, and evaluation of methods. The knowledge is then applied to the domain of crime scene analysis, with a focus on detecting serial residential burglaries. This thesis introduces a procedure for systematic collection of crime scene information. The thesis also investigates impact and relationship between crime scene characteristics and how to evaluate the descriptive model results.

The results suggest that the use of descriptive and predictive models can provide feedback for crime scene analysis that allows a more effective use of law enforcement resources. Using descriptive models based on crime characteristics, including Modus Operandi, allows law enforcement agents to filter cases intelligently. Further, by estimating the link probability between cases, law enforcement agents can focus on cases with higher link likelihood. This would allow a more effective use of law enforcement resources, potentially allowing an increase in clear-up rates.