Master of Science in Computer Science
February 2017

Linking Residential Burglaries using the Series Finder Algorithm in a Swedish Context

Aleksandr Polescuk

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science of Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Aleksandr Polescuk
E-mail: alpo14@student.bth.se

University advisor:
Dr. Martin Boldt
Department of Computer Science and Engineering

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Context. A minority of criminals performs a majority of the crimes today. It is known that every criminal or group of offenders to some extent have a particular pattern (modus operandi) how crime is performed. Therefore, computers' computational power can be employed to discover crimes that have the same model and possibly are carried out by the same criminal. The goal of this thesis was to apply the existing Series Finder algorithm to a feature-rich dataset containing data about Swedish residential burglaries.
Objectives. The following objectives were achieved to complete this thesis: Modifications performed on an existing Series Finder implementation to fit the Swedish police forces dataset and MatLab code converted to Python. Furthermore, experiment setup designed with appropriate metrics and statistical tests. Finally, modified Series Finder implementation's evaluation performed against both Spatial-Temporal and Random models.
Methods. The experimental methodology was chosen in order to achieve the objectives. An initial experiment was performed to find right parameters to use for main experiments. Afterward, a proper investigation with dependent and independent variables was conducted.
Results. After the metrics calculations and the statistical tests applications, the accurate picture revealed how each model performed. Series Finder showed better performance than a Random model. However, it had lower performance than the Spatial-Temporal model. The possible causes of one model performing better than another are discussed in analysis and discussion section.
Conclusions. After completing objectives and answering research questions, it could be clearly seen how the Series Finder implementation performed against other models. Despite its low performance, Series Finder still showed potential, as presented in future work.

Keywords: Crime linkage, Modus Operandi, Series Finder, Residential Burglaries.

Acknowledgements

I would like to thank my thesis supervisor Dr. Martin Boldt of the Department of Computer Science and Engineering at Blekinge Institute of Technology. Whenever I ran into problems while writing this thesis, I could always turn to Prof. Boldt for help. He guided me throughout this long and exciting research journey step by step. Without Dr. Martin Boldt, this research would not have been possible as his input was vital. Every question or concern was quickly answered when needed.

I would also like to thank Dr. Anton Borg of the Department of Computer Science and Engineering at Blekinge Institute of Technology. The initial and ongoing discussions were essential on how to modify the original implementation. The feature mapping suggestions were crucial as well.

I would like to acknowledge my colleague Mattias Albinson for proofreading and providing valuable comments on this thesis.

Finally, I would like to express my gratefulness to my beloved ones for never-ending support and motivation throughout my studies and this thesis. This achievement would not have been possible without them.

Thank You!

Aleksandr Polescuk

Contents

Abbreviations

ANOVA – Analysis of Variance
AUC – Area Under Curve
CD – Critical Difference
CPM – Crime Prediction Model
DS – Degree of Dynamics
DTW – Dynamic Time Wrapping
M.O. – Modus Operandi
RECAP – Regional eCrime Analysis Program
SF – Series Finder
SMMSM – Segmented Multiple Metric Similarity Measurements
SQL – Structured Query Language

List of Figures

List of Tables

List of Equations

Chapter 1

Introduction

1.1 Introduction to problem area

In today's world, increasing crime rate is a huge challenge [1][2][3]. In order to decrease it, appropriate means have to be taken [2]. One of such means can be applications of computers' computational power. Not so long ago all details about crimes were recorded only by humans. However, this method was prone to human errors and led to inaccurate or insufficient information about the case, therefore, aggravating investigation processes [4]. Not mentioning the time required to re-read all recorded data. As the amount of data is considerably increasing, humans can no longer handle it efficiently by themselves. Thus, computers, as an aid are being used to handle it more effectively. Data analysts have increased the overall throughput of the crime investigation process by applying machine learning and data mining. These are part of artificial intelligence, which uses different algorithms to learn distinct concepts [4][5]. The goal of such approach is an adaptation to data growth without any intervention. With data increase, higher accuracy from pattern detection techniques is expected. As nowadays investigation process takes into account many different and specific details, it becomes barely possible for the analysts to investigate the case manually and efficiently at the same time [6]. Therefore, it is crucial to keep exploring the current research area. With more research in the field, newer, more advanced techniques are being investigated, thus increasing chances to invent suitable system to diminish criminal world [7].

Research by Nath has shown that roughly 50% of crimes are committed by about 10% of the criminals [4]. Knowing such statistics, investigating various pattern detection techniques becomes significantly important. Repetitive crimes are performed with some model or a plan. If the pattern is known, the offender can be stopped before another crime takes place [8]. Furthermore, old unsolved crimes can be revealed, if there are various algorithms, techniques, and models proposed for crime pattern detection [4][5][8]. Most available methods apply semi-supervised learning [4]. Aside from those, some of the modified traditional clustering methods can perform accurately enough [9]. Such tools are used to enhance the process of investigation, not to automate it. As offenders' modus

operandi (M.O.) are not static and are evolving over time, fully automated processes are much more complicated and not always feasible [2]. Therefore, the combination of both experienced data analysts and semi-automated techniques can become a robust solution for detecting the evolving crime patterns.

Particularly for this research, a novel pattern detection algorithm called Series Finder (SF) was chosen. Series Finder algorithm mimics the same method used by analysts. It uses databases and searches for any similarities between selected attributes, thus trying to identify M.O. [8]. The more crime data available, the better outcome is achieved. Each crime can have very different M.O. including, for instance, the type of crime, chosen tools used, time and place to name a few [8]. The more relevant details available about the crime, the better M.O. is captured. Correct and detailed knowledge about the M.O. increases the chances to apprehend the criminals. Definition of M.O. can be equated to the style of the criminal. As every criminal has some degree of uniqueness in the M.O. it is possible to classify unsolved crimes into various clusters based on M.O. details [9]. The more correctly unsolved cases are clustered, the higher chance to solve them.

In this study, the aforementioned pattern detection algorithm Series Finder was used with real burglary data obtained from Swedish police. Modifications of the algorithm were required to fit the given data. As a contribution, the performance of the Series Finder algorithm was evaluated and compared to a state-of-the-art algorithm, i.e. Spatial-Temporal. In addition, Series Finder was evaluated against baseline comparative, i.e. Random model. Despite the fact that implementation was modified according to Swedish police dataset, it does not mean that this project is only relevant in a Swedish context. Techniques investigated in this study could probably be applied in most national police forces.

1.2 Related work

There are many techniques, models, and algorithms proposed for crime pattern detection.

Chandra et al. have proposed multivariate time series clustering approach to detect different crimes [10]. Such method required dynamic time wrapping as Euclidean distance based measuring was mostly used for non-time series and was not suitable for a proposed multivariate clustering system [10]. Dynamic time wrapping (DTW) had a drawback while working with varying weights, so a Minkowski model was proposed to overcome such limitation. The proposed approach performed well on dimensions with different weights of crimes [10].

Another approach was demonstrated and proposed by Babakura et al. [11]. They used modified classifiers to predict crime patterns. One classifier was Naïve Bayes while another was Neural Networks using Back Propagation of errors. They also used two different datasets to evaluate the classifiers mentioned above. Naïve

Bayes algorithm outperformed Back Propagation on both datasets. However, overall both algorithms had high accuracy. Their research proved that machine learning and data mining could contribute to a great extent if used by law enforcement agencies [11].

The use of clustering techniques as a method to detect crime patterns was proposed by Nath [4]. This technique referred to semi-supervised or expert-based learning. It required data analysts to work together with detectives to choose the most important attributes. Clustering techniques allowed the application of different weights of attributes based on crime type and to which cluster it was assigned. Furthermore, it allowed weighting of not only numerical but also categorical attributes [4], making this approach more flexible. The proposed method was able to detect crime patterns from a large dataset, thus facilitating work for crime specialists.

Munasinghe et al. proposed a mixed method approach [5]. First of all, they extracted most relevant data from police crime records and then converted it to use in quantitative analysis. Furthermore, a M.O. extraction form was used to derive better defined M.O., as it was vital while searching for crime patterns [5]. Afterward, two clustering algorithms were evaluated with pre-processed data. The whole approach had shown good overall results at finding crime patterns. However, it had some limitations such as applicability, but with some adjustments, those limitations could be addressed [5].

Ozgul et al. used the crime prediction model (CPM) algorithm [12]. CPM sorted dataset into solved and unsolved clusters, which were further investigated to extract similarities between crimes. CPM precision was high, whereas recall values were low for larger terrorist groups. However, on smaller terrorist groups, the recall values were greater. Primary attributes which were taken into consideration were M.O., date, and location [12]. Authors planned to further investigate such approach because of the positive results.

A different classification system was proposed by Dahbur et al. [13]. The approach was based on a cascaded network of Kohonen neural networks which were backed up with heuristics. According to the authors, their proposed project was mainly applied to armed robberies. Nevertheless, it could be used in for other types of crimes, but with some modifications [13]. The system could be adjusted by every investigator particularly, thus making it flexible. The overall results were promising. However, the qualitative analysis revealed that system's accuracy was only about 64%. Further experiments were planned to increase the accuracy of the system. Despite the present results, it could still be used as an aid by criminal analysts [13].

Brown et al. introduced a data association method to assist crime analysts like a decision support system [14]. The suggested decision support system reduced the required time significantly for searching criminal patterns throughout the databases. Furthermore, the accuracy was equal to experienced crime analysts. The system was based on structured query language (SQL). It was also compared

to similar techniques applied at that time and successfully outperformed by time and accuracy [14]. Further work was planned for additional improvements.

Buczak et al. applied fuzzy association rules to establish patterns of similar crimes [6]. According to authors, the proposed approach facilitated in searching for most relevant crimes while skipping unrelated ones. Furthermore, the particular rules could be applied to a national level, thus perpetrating it more attractive for potential users. The proposed technique performed quite well with the reduction of total amount of rules by 95.2 %, thus leaving 675 rules in total for the crime analysts. However, the results revealed that further improvements and analysis were required to gain improved insight into crime patterns [6].

The framework called Regional eCrime Analysis Program (RECAP) to identify criminals was proposed by Brown [15]. The system consisted mainly of two parts: data fusion – preparation of data for later use and data mining – to find similar patterns of crimes in the data. RECAP provided methods for temporal, spatial and attribute matching methods, thus enlightening the whole investigation process for law associates [15]. The system's ability to filter out and display related cases saved a lot of time and effort for crime analysts.

Wang et al. proposed an association model which was based on M.O. [16]. The model used M.O. to establish a connection between different links in the dataset. The introduced model was used on two different datasets: robbery and residential burglary. Despite positive results in overall, the model performed better on robbery than residential burglary crime records. Consequently, further work was required to tune the current model [16].

Chunyu et al. presented an improved frequent predicate algorithm to facilitate the work of law enforcement agencies [17]. The approach incorporated multidimensional association rules which depended on the minimum support and minimum confidence parameters. The algorithm was tested using g-statistics. The overall results were reliable, but the approach required further investigation and testing.

The clustering technique was introduced to the crime field by Zhou et al. [9]. Authors stated that basic clustering methods were not accurate enough to meet the requirements of law enforcement agencies. Therefore, they introduced hybrid similarity measurement called the segmented multiple metric similarity measures (SMMSM). The proposed approach incorporated different measurements to find similarities between chosen objects. The outcome of such technique was positive as it outperformed traditional clustering methods by accuracy and efficiency [9].

To sum up, it can be seen that many different techniques exist that contribute to the process of detection of similar patterns between objects. Nevertheless, none of the solutions showed performance on such a level that it conquered the field. Most of the techniques brought promising results, however, at the same time, many of them required extra work in order to be finished and to improve performance. Despite future work needed, some solutions were favorable enough to serve as an aid tool in crime pattern detection.

1.3 Problem statement

After analysis of the related work, the following research gap has been identified:

The existing Series Finder implementation was not evaluated against a state-of-the-art algorithm. Thus, its real potential is unknown. Furthermore, it was not modified to fit a feature-rich dataset such as the one collected by the Swedish police. Therefore, the research described in this thesis was conducted to cover the mentioned gap.

1.4 Aims and objectives

The aim of the research was to find patterns of serial crimes (i.e. crimes committed by the same burglar(s) that had the same modus operandi signature) using a modified version of the Series Finder algorithm on the real burglary data obtained from Swedish police. In order to reach the aim, the following objectives were defined:

- To modify the existing Series Finder implementation to fit the Swedish police data and convert it from MatLab to Python code.
- To design an experiment setup with suitable dataset, metrics, statistical tests and conclusions.
- To evaluate Series Finder implementation compared to a state-of-the-art algorithm and a baseline model.

1.5 Research questions

In order to complete the objectives, the following two research questions were investigated in this thesis.

RQ1. How can Series Finder be implemented in a Swedish context?
Motivation: It is important to choose the appropriate method for modification. Flexibility in modification is a desired advantage, leading to less time and effort required. Most of the time, datasets have different features and distribution, thus requiring modification of existing implementation.

RQ2. How accurate is the modified implementation of Series Finder compared to a state-of-the-art algorithm?
Motivation: In order to find out the potential of modified version of Series Finder, it has to be evaluated against a known model. Furthermore, it is important to compare different measurements of comparable implementations.

1.6 Limitations

Originally, seeds consisting of two crimes had to be provided to run the Series Finder implementation. However, as most of the linked series had only two crimes it was decided to use one crime as a seed. In addition, the original implementation had poor documentation, leading to many obstacles to make it work with two crimes as a seed. A further issue was that Series Finder took too long to run. To reduce the time required, the data sample was limited. With increased data sample size, execution time increased exponentially. Another issue was that the 32-bit Python environment implicitly chosen by the original developers of Series Finder did not provide enough memory to run the Series Finder with a larger dataset. Lastly, limited MatLab knowledge led to more time required to understand the code.

1.7 Method used

In the proposed research, the experimental methodology was chosen to find answers to the research questions. Experimental research is a scientific approach to investigate how research can manipulate controls, variables and their changes throughout the process [18]. The following steps were identified and performed prior to and during the experiment, in order to achieve valuable results.

1. Modification of existing Series Finder implementation to fit the Swedish police residential burglary data. The original code was written in a scripting language called MatLab. The original implementation had to be translated and modified into the Python programming language.

2. The modified Series Finder implementation was evaluated compared to state-of-the-art algorithms. The chosen rivals for Series Finder implementation was a Spatial-Temporal model based on space and time distances and Random model as a baseline.

3. The aforementioned implementations were evaluated by comparing the following metrics: mean accuracy, recall, precision, AUC, error rate and F-score.

4. The proposed experiment's independent variable was rather "linkage method-/algorithm" which has three different levels: Series Finder, Spatial-Temporal and Random. All metrics and results were dependent variables and might have changed according to modifications done on the algorithms.

5. After all metrics and results were obtained, statistical tests were used to show how each algorithm performed. The Kruskal-Wallis test was used to find differences between the three algorithms. If differences existed, the

Nemenyi post-hoc test was applied to distinguish between the algorithms. Besides, Cohen's d accompanied statistical tests to quantify the effect sizes.

6. Depending on the statistical analysis results, further modifications could be done to achieve better overall performance.

1.8 Thesis structure

In this section, the structure of the thesis is briefly introduced.

Introduction – presents the research area, followed by related work revealing which methods have previously been applied to detect similar crime patterns. Afterward, the problem gap is identified. From the gap, aims and objectives together with research questions are formed. Finally, the chosen methodology is described.

Background – introduces relevant concepts present in the selected field of research. Notable metrics and tests are presented together with their importance. The dataset used is described in detail, followed by descriptions of the Series Finder algorithm, Spatial-Temporal model and Random model.

Method – describes the selected approach and its details to reach the objectives. The feature mapping, the changes made to Series Finder and how the experiment is performed are explained.

Results – reveals the results gathered from the selected metrics and statistical tests.

Analysis and discussions – presents the answers to the research questions and the analysis of the results. Furthermore, possible factors of impact on each model's results are discussed.

Conclusions and future work – presents the conclusions and possible future work.

Chapter 2

Background

2.1 Key concepts

Modus Operandi (M.O.) is Latin and stands for "method of operation" [19]. Most often this term is used by national police forces to describe how criminals behave during the crime [5][20]. Furthermore, M.O. is often used as an item of analysis by methods for detecting similarity pattern between crimes [8]. It tends to change gradually over time, but as the variance is small, it is considered as a reliable measurement [20]. Additionally, M.O. distinguishes one or another possible criminal for the current crime.

According to Bruce et al. **crime patterns** can be defined when two or more crimes are grouped together based on similarity of specific conditions [21]. Some of the specific conditions could be, i.e. perpetrator's entrance behavior, Spatio-temporal data (geographic and parameters) and type of stolen goods. The specific conditions often fall under the M.O. definition.

Features or attributes – in this thesis, both expressions stand for characteristics of the particular crime.

Instance - may be called a particular crime from the dataset. An instance is to be classified by the Series Finder algorithm.

Classification - is a concept from the field of machine learning which predicts instances' class depending on its attributes/features, i.e. characteristics [22]. One of the simplest classification techniques is binary classification, which has only two values to classify [22][23].

Known linked crimes – are the crime instances that are related to particular crime pattern in the dataset. These crime examples are linked based on physical evidence, i.e. DNA or fingerprints [20]. Therefore, they can be used as a measurement for comparison and classification of the estimated linked crimes, i.e. ground truth.

Estimated linked crimes – particularly in this thesis, it represents crimes identified by algorithms to be carried out by ordinary criminals potentially. Therefore, such instances can be related to the same pattern which is for known linked crimes. This would mean that a new instance has same crime pattern and belongs to the same criminal or group of offenders [20].

2.2 Metrics and their importance

Appropriate metrics have to be used to identify differences or similarities between particular elements. In our case, we want to establish the contrast between the algorithms. Therefore, the following metrics are chosen:

- Accuracy – defined as the overall correctness of the classifier or as the amount of correctly classified instances [22][24]. Accuracy is determined by the formula (2.1):

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{2.1}$$

where tp – true positives which stands for correctly classified positives, i.e. correctly identified linked crimes; tn – true negatives which stands for correctly classified negatives, i.e. correctly identified non-linked crimes; fn – false negatives which stands for incorrectly classified positives, i.e. incorrectly classified crimes belonging to linked series; fp – false positives which stands for incorrectly classified negatives, i.e. false crimes classified as belonging to linked series [22].

- Recall – also known as sensitivity and defined as the true positive rate [22][24]. It can also be defined as the ratio of correctly classified instances to a total number of correct instances [25]. The recall is determined by the formula (2.2):

$$Recall = \frac{tp}{tp + fn} \tag{2.2}$$

- Precision – also known as positive predictive value and is defined as the fraction of correctly classified instances, divided by the true positives and false positives [24][25]. Precision is determined by the formula (2.3):

$$Precision = \frac{tp}{tp + fp} \tag{2.3}$$

- The area under the curve (AUC) – also known as the c-statistic and is defined according to formula (2.4) [24]. AUC is a measure of the discriminatory power of a predictive model and is referred to as the balanced accuracy:

$$AUC = \frac{1}{2}\left(\frac{tp}{tp+fn} + \frac{tn}{tn+fp}\right) \tag{2.4}$$

- F-score – defined as the measure of a test's accuracy which uses precision and recall values for estimation [24]. F-score is determined by the formula (2.5):

$$F\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.5}$$

- Error rate – defined as the fraction of incorrectly classified instances compared to the total amount [22]. The error rate is determined by the formula (2.6):

$$Error\ rate = \frac{fp+fn}{tp+fn+fp+tn} \tag{2.6}$$

The preceding metrics are essential to gain superior insight while evaluating different models or algorithms.

2.3 Statistical tests

Kruskal-Wallis test

Kruskal-Wallis test is a non-parametric statistical test for identifying whether selected elements belong to same distribution [26][27]. Mostly, this test is used as an alternative to the parametric one-way analysis of variance (ANOVA) [26]. The primary purpose of this test is to determine that at least one case stochastically dominates the other cases. However, Kruskal-Wallis test does not reveal the exact part where the stochastic dominance appears [26]. In order to detect stochastic dominance explicitly, other tests can be applied, such as Dunn's test or Nemenyi test. In addition, the Kruskal-Wallis test does not require normal distribution or equal variance among evaluated models, thus facilitating the whole process [28]. The Kruskal-Wallis test is performed according to formula (2.7):

$$G = \left[\frac{12}{N(N-1)}\sum_{j=1}^{k}\frac{R_j^2}{n_j}\right] - 3(N+1) \tag{2.7}$$

where N – the total number of samples in all cases; n_j – the number of samples in case j; R_j – the rank of the case j; and k – the number of samples in one case [26].

Nemenyi post-hoc test

The Nemenyi post-hoc test is defined as a tool for testing the significance of the differences between candidates [29]. It is similar to Tukey test for ANOVA and is mostly used when two or more candidates, either algorithms or classifiers are compared with each other [7]. The Nemenyi test takes the performance of comparable algorithms into consideration. Algorithms are considered to differ significantly if their mean ranks differ by at least the critical difference (CD) [29]. Based on the CD, candidates are evaluated and grounded by their rank which depends on their performance. The experiment done by Demsar reveals that more accurate results can be obtained while comparing other algorithms to one control algorithm rather than pair-wise [7]. However, this is relevant only if more than two algorithms are compared with several datasets. CD is determined using formula (2.8):

$$CD = \sqrt[q]{\frac{K(K+1)}{6D}} \tag{2.8}$$

where q – stands for critical value, K – stands for a number of conditions, D – stands for a number of participants.

In order to determine how different algorithms or models perform, Kruskal–Wallis and Nemenyi post-hoc tests are applied. After application of both tests, conclusions about performance and algorithm differences can be done.

Effect size (Cohen's d)

The Cohen's d is known as effect size and is often used to calculate the standardised difference between two means when comparing different models [30][31]. Normally it can be found as a complement to statistical tests, such as ANOVA, Nemenyi or T-test. Cohen's d can be determined using formula (2.9) [30]:

$$Cohen'sd = \frac{S_1 - S_2}{SD_{pooled}} \tag{2.9}$$

where S_1 and S_2 – the means of sample 1 and sample 2, SD_{pooled} – the standard deviation of samples S_1 and S_2 and is determined using formula (2.10):

$$SD_{pooled} = \sqrt{\frac{SD_1^2 + SD_2^2}{2}} \tag{2.10}$$

where SD_1 and SD_2 stands for standard deviation. Effect size value is usually between 0.01 and 2.0 [32]. Furthermore, effect size most often contributes when searching for statistical significance.

2.4 Swedish police dataset description

Datasets can consist of different types of data. To be successful in the experiment, it is essential to know with what kind of data we have to deal. Different types of data have to be handled in a particular way. The original dataset obtained from the Swedish police force is in SQL format and contain 140 different features, i.e. a feature-rich dataset compared to existing burglary datasets. The dataset has 8155 instances of residential burglaries. Especially for this experiment, the newest crime instances and most relevant features are used. Therefore, crimes from 2015 are chosen with 41 most relevant features. The dataset consists of two parts. The first portion of the dataset with most relevant features can be seen in Table 2.1.

Feature	**Format**	**Comment**
id	integer	crime position in data sample
idval	string	crime identification code
datestart	string	date when the crime started
timestart	string	time when the crime started
dateend	string	date when the crime ended
timeend	string	time when the crime ended
longitude	float	coordinate where the crime happened
latitude	float	coordinate where the crime happened

Table 2.1: First part of the Swedish dataset

The second part consist of binary features, i.e. 1 or 0, representing whether a certain feature is present at the burglary crime scene or not. The features are shown in Table 2.2.

Features		
basement	cellardoor	door
sect62unknown	sect65unknown	balconydoor
sect65other	mailslot	abovegroundlvl
groundlvl	noviewcover	triplepanewindow
toolfromplace	breaks	breakswindowin
ventpos	drills	sect64unknown
unlocked	aptrental	aptowned
villa	farm	plannedabsence
patiodoor	mirrorpatiodoor	window
nobpu	illegalkey	sect64other
prevbreakin	townhouse	spontabsence

Table 2.2: Second part of Swedish police dataset

2.5 Dataset used in original Series Finder implementation

The original Series Finder dataset is different compared to Swedish police dataset as it contains fewer features. Despite this fact, the current features that are used are more detailed. E.g location of entry: door dispersed into door - basement, door - front, door - rear, door - side, door - sliding and door - unknown. The dataset used in this thesis does not have such detailed features. The original implementation is lacking a description of how the features are applied. Thus, they can not be modified to fit the performed experiments. Therefore, combinations of features are used to fit implementation as much as possible. In Table 2.3 it can be seen what the algorithm requires as a location of entry information. There are other important features, such as means of entry and premises, but they are not described in detail here. The format of all features is the string.

LocEntry features		
null	Basement	Door: Basement
Door: Rear	Door: Side	Door: Sliding Glass
NA	Roof	Skylight
Wall	Window: Basement	Window: Fire Escape
Window: Ground	drills	Window: Ladder
Window: Side	Window: Skylight	Window: Unknown
Door: Front	Door: Unknown	Window: Front
Window: Rear	Unknown	

Table 2.3: Location of entry features of original Series Finder dataset

There are more features available such as suspect's race, height, weight, hair color, ethnicity, and gender. However, as the Swedish police dataset has no such features, they are not taken into account when evaluating the algorithm.

2.6 Series Finder algorithm

In order to detect patterns of crime, Series Finder algorithm operates under similar to the supervised learning method [3][8].

To detect patterns, Series Finder manipulates two particular types of weights: the pattern-specific and the pattern-general weights [3]. As the pattern advances, the specific-pattern weights are modified, thus making possible to capture similar M.O. The general-pattern weights are used to find similar characteristics of all available patterns [8].

In general, we can start describing a model with the following expression:

$P = \{C_1, C_2, \ldots, C_n\}$, where P – pattern of crime; C – crime instance; At first, only a few instances are known from the P set. Afterward, a few known instances are used by Series Finder to generate a set of D – the similar crimes. Then Series Finder endeavor set D to become similar to the set P, which at the beginning is unknown. In order to add sequentially new instances to the set D, new elements are chosen from candidate crime set C_D [8]. Set C_D stands for crimes that happened in the same time lapse as crimes from the P set.

The following Series Finder implementation is divided into two parts: crime-crime similarity and pattern-crime similarity [8].

Crime-crime similarity: The crime-crime similarity part β_D checks for similarities between C_i and C_k crime instances throughout a set of similar crimes D. The process is defined by formula (2.11) [8]:

$$\beta_D(C_i, C_k) = \sum_{j=1}^{J} \omega_j \varphi_{D,j} \, S_j(C_i, C_k) \tag{2.11}$$

where ω_j– are pattern-general weights; $\varphi_{D,j}$– are pattern-specific weights; J– is the similarity measure between crime instances; S_j– represents the similarity of the j^{th} attribute. To obtain high β – the comparable crime instances must have common similarities among chosen attributes in the particular crime pattern as well as generally to all patterns [8].

The pattern-specific weights are defined by formula (2.12) [8]:

$$\varphi_{D,j} := \frac{1}{\Gamma_D} \frac{1}{|D|(|D|-1)/2} \sum_{j=1}^{|D|} \sum_{k=1}^{|D|} S_j(C_i, C_k) \tag{2.12}$$

where Γ_D – a normalizing factor, calculated according to formula (2.13), D – similar crimes set:

$$\Gamma_D = \sum_{j=1}^{J} \varphi_{D,j} \tag{2.13}$$

The pattern-general weights are used to generalize how essential the selected attributes are. Afterward, the chosen attributes are trained on the past patterns which were characterized by the analysts. Therefore, a gain function that incorporates balance into recall and precision values is used. Expressions are determined in (2.14) and (2.15) [8]:

$$Recall(P, D) = \frac{\sum_{C \in P} 1(C \in P)}{|P|} \tag{2.14}$$

$$Precision(P, D) = \frac{\sum_{C \in D} 1(C \in P)}{|D|} \tag{2.15}$$

where P – true pattern, D – similar crimes set. Based on the recall and precision the gain function G is calculated according to formula (2.16):

$$G(D, P, \alpha) = Recall(P, D) + \tau \times Precision(P, D) \tag{2.16}$$

where τ – a trade-off weight between two essential characteristics of the instances; α – used to maximize the gain over the all patterns in the training set [3].

Pattern-crime similarity: The pattern-crime similarity part is responsible for checking whether crime instance i is identical enough to be likely included into set D. Therefore, dynamism is integrated into M.O. The pattern-similarity is determined by formula (2.17) [8]:

$$S(D, C) := \left(\frac{1}{|D|} \sum_{n=1}^{|D|} \beta_D(C, C_n)^d \right)^{(1/d)} \tag{2.17}$$

where S – is the similarity between crime instance C and set D; where $d \geq 1$ and is defined as a degree of dynamism; where $C \in C_D$ and $C_n \in D$;

Invoking a soft-max function allows the set of patterns D to derive. If the size of d is large, it is easier for a crime instance to be considered as similar to the set of D. On the other hand, if d is small, the pattern becomes stable, and the crime

instance has to be very similar to be included into the set D. Therefore, variable d allows balancing between strict and loose patterns [3][8] .

Series Finder algorithm: The algorithm works in the following way: It starts with just a few crime patterns and then crime instances are recursively moved from C_D set to the D set. Then it recursively iterates through all crime instances with highest pattern-crime similarity compared to D, and adds them to set D. Afterward, $D's$ cohesion is evaluated using formula (2.18) :

$$Cohesion(D) = \frac{1}{|D|} \sum_{C_n \in D} S(D/\{C_n\}, C_n) \tag{2.18}$$

where D – is similar crimes set; S – is similarity measure; C_n – is crime instance.

The following is the formal Series Finder algorithm [8].

1: Initialization: $D \leftarrow \{Seed\ crimes\}$
2: Repeat
3: $\quad C_{tentative} \in \arg\max_{C \in (C_D \setminus D)} S(D, C)$
4: $\quad D \leftarrow D \cup \{C_{tentative}\}$
5: $\quad$ Update $\varphi_{D,j}$ for $j \in \{1, 2, \ldots J\}$, and Cohesion(D)
6: Until Cohesion(D) $<$ threshold
7: $D^{final} := D \setminus C_{tentative}$
8: Return D^{final}

where *threshold* – is a boundary for growing the D set.

This pseudo-code can be used to implement the Series Finder algorithm in the chosen programming language.

2.7 Spatial-Temporal model

Spatial-Temporal model is another possible approach for crime pattern detection [33]. This model performs on information obtained from space (where a crime occurred) and time (when a crime occurred). Both space and time measurements are equally important. Therefore, they affect the final result equally. Implementation takes longitude, latitude, date and time of one crime as input and returns most similar crimes according to space and time. As both values are equally weighted, a summarized value is concluded to specify the closeness of the crimes.

On the one hand, this model can be efficient, but on the contrary, performance may significantly decrease according to the chosen parameters. The performance

of this model is very dependent on the dataset used. Another thing to consider is a number of crimes to search for. As on some occasions, searching for fewer crimes may lead to better performance than trying to find as many as possible. Besides, if the implementation will return too many possible crimes related to the particular linked series, the overall performance may decrease at the same time increasing the manual work for police forces rapidly. Therefore, to use this model, trade-off shall be taken into account. The similarity for crime pattern detection can be calculated using formula (2.19)[33].

$$ST = \beta \times S_m + (1 - \beta) \times T_m \tag{2.19}$$

where ST – stands for Spatial-Temporal, β – trade-off coefficient between space and time, S_m – space measurement (in km between crime-pairs) and T_m – time measurement (in hours between crime-pairs).

2.8 Random model

Random model is essentially used as a baseline to compare one or more different models [34]. This model provides useful information, such as whether the comparable model is at least better than randomly generated results. It may be implemented in many different ways. Therefore, the implementation itself is dependent on the experiment requirements. Notwithstanding frequently poor performance of this model, it provides significant insight into the results. The initial results of this model can reveal whether it is profitable or worthy to pursue the experimentation further. In this paper, Random model is built in the specific way to fit the experiment. There is no particular formula for a Random model. Further, the description of how a model works is presented:

For 500 given crimes, this model randomly selects ten other crimes from the dataset using a uniform distribution. Therefore, its performance most of the time is low. With the increase of dataset size, model's performance goes down. There is a trade-off when choosing sample size and how many crimes to search for.

Chapter 3

Method

In this thesis, an experimental methodology was chosen in order to achieve objectives. Other methods such as interviews, surveys or case studies were not applicable to this thesis as they can not measure classification performance of algorithms, which is one of the objectives of this thesis. In the following sections, most important parts of this experiment are discussed in detail.

3.1 Selected features in Swedish police dataset

As discussed before, the Swedish police dataset was different compared to the dataset used in Series Finder implementation. Therefore, the first objective was to map the features of Swedish police dataset to the Series Finder implementation dataset. The most important features to map were the binary ones, like date, time, coordinates and some more were already matched. Despite how feature-rich the Swedish police dataset was, most of the features could not be used because of the Series Finder's origins. To fit the Series Finder implementation as much as possible, the dataset's binary features were joined into logic expressions. Some of the examples are provided in Table 3.1.

Series Finder features	**Swedish police dataset features**
Door: Basement	basement AND (patiodoor OR mirrorpatiodoor OR balconydoor OR cellardoor OR door)
Window: Front	(window OR triplepanewindow) AND noviewcover
Broke Glass	breakswindowin
Apartment	aptrental
Condominium	aptowned

Table 3.1: Feature mapping

Many other features were mapped either in a similar way or one to one. There were occasions where features could not be mapped. Therefore, they were disabled and not used. This way, internal calculations of the Series Finder were untouched, keeping it as stable as possible.

3.2 Modifications of Series Finder

The Series Finder implementation was initially written in the MatLab scripting language. For this project, it was converted to Python programming language. That is a language which law enforcement in Sweden prefers to implement their analysis methods in. In addition, it is open-source, free and has many different libraries.

The original, Series Finder implementation gathered the following computations before running the main algorithm:

- Coordinates - checked for similarities in distance between crimes.
- Locations of entry - checked for similarities in where the crime happened.
- Means of entry - checked for similarities on how the crime happened.
- Days apart - checked for similarities in when the crime happened.
- Day of the week - checked for similarities on which day of the week crime happened.
- Premises - checked for similarities in type of premises where the crime happened.
- Ransacked - checked for similarities whether premises were ransacked.
- Residents - checked for similarities if residents were present at home while crime happened.
- Suspect - checked for similarities on how suspect looks.
- Victim - checked for similarities on how victim looks.
- Timeframe - checked for similarities in time window when crimes happened.

Swedish police dataset did not have all essential features. Therefore, the following parts were removed:

- Suspect - it has been removed because there was no information about suspects in our dataset.
- Victim - it has been removed for the same reason as suspect part - no information available.
- Timeframe - it has been removed because it was not always present in a dataset and imprecise.

Another modification which was done was the seed size to be provided to the Series Finder implementation. Originally it required at least two crimes as a seed, but because of our data from Swedish police, we had to reduce it to one. The reason was that most of the linked series provided by the Swedish police were only made up of two crimes. In order to be able to verify that the implementation worked, we needed to have at least one crime to search for.

3.3 Experimental approach

In order to achieve our objectives, an experiment was performed. The Series Finder implementation had many parameters which could be tuned. The most important ones were cutoff, a degree of dynamics and maxlen. The cutoff value determined how similar the crime had to be to get added to the list of possible crimes belonging to the same pattern. The degree of dynamics determined how flexible the pattern could be. Maxlen determined the maximum amount of crimes that could be found. The following metrics were calculated to gain information about the performance of different models:

- Recall
- AUC
- F-score
- Precision
- Accuracy
- Error rate

When all the metrics were calculated, the following statistical tests were applied:

- Kruskal-Wallis
- Nemenyi post-hoc

As a complement to the statistical tests, Cohen's d was additionally calculated.

The experiment was performed with 12 linked series. In Table 3.2 we can see the distribution of the linked series.

Linked series	**Series size**
Series1	2 crimes
Series2	2 crimes
Series3	3 crimes
Series4	2 crimes
Series5	2 crimes
Series6	2 crimes
Series7	2 crimes
Series8	2 crimes
Series9	4 crimes
Series10	2 crimes
Series11	4 crimes
Series12	2 crimes

Table 3.2: Number of crimes in each linked series

3.3.1 Data preprocessing

In order to fit the pythonic version of the Series Finder, the data sample had to be preprocessed in a specific way. Few scripts were written to keep some of the data in integer or string formats where needed. The sample size of 500 was chosen because of several causes. First of all, the environment had limited the possibility to run the implementation with more samples, and another issue was the execution time. Therefore, to make it most optimal to run - 500 samples were chosen. Every run, X amount of samples were randomly selected from the data. X can be defined as the size of the linked series subtracted from 500. The crimes belonging to currently investigated series were added to the data sample. Afterward, the sample was checked for duplicates, and if any were found, they were removed and replaced with another crime so that the total count always was 500 samples. This way, the sample size was kept fixed for every experimental run and randomized.

3.3.2 Initial experiment

In order to find the right cutoff and degree of dynamics (ds) values for Series Finder for the main experiment, an initial experiment was performed. The experiment contained five different cutoff values and three different the degree of dynamics values. The experiment was carried out with every crime as a seed. Therefore, 29 outputs were gathered in total, with average value concluded from them. In Table 3.2 can be seen 12 linked series with 2, 3 or 4 crimes in it, leading to a total sum of 29 crimes over the all linked series. With each cutoff and ds parameter, three runs were made. The total amount of runs added to 45 per 1 crime. Recall and AUC values were most important as they revealed the actual performance of the implementation in this experiment. Thus, these metrics were chosen as deciding factors for identifying the most optimal cutoff and ds values. Also, F-score value was taken into account, but it did not heavily affect the final result. In Tables 3.3, 3.4 and 3.5 we can see the outcome of initial experiment.

Recall	**ds1**	**ds3**	**ds5**
cutoff 0.01	0.055	0.225	0.106
cutoff 0.10	0.027	0.132	0.018
cutoff 0.19	9.26 x 10^{-3}	0.116	0.025
cutoff 0.28	0.00	0.039	9.26 x 10^{-3}
cutoff 0.37	6.94 x 10^{-3}	0.032	0.016

Table 3.3: Recall metric. Cutoff between 0.01 and 0.37 with a stepsize of 0.09 and degree of dynamics (ds) between 1 and 5 with a stepsize of 2.

AUC	**ds1**	**ds3**	**ds5**
cutoff 0.01	0.522	0.603	0.544
cutoff 0.10	0.511	0.562	0.505
cutoff 0.19	0.502	0.555	0.510
cutoff 0.28	0.498	0.517	0.502
cutoff 0.37	0.501	0.514	0.506

Table 3.4: AUC metric. Cutoff between 0.01 and 0.37 with a stepsize of 0.09 and degree of dynamics (ds) between 1 and 5 with a stepsize of 2.

F-score	**ds1**	**ds3**	**ds5**
cutoff 0.01	0.018	0.043	0.023
cutoff 0.10	0.019	0.059	9.52 x 10^{-3}
cutoff 0.19	9.26 x10^{-3}	0.063	0.015
cutoff 0.28	0.00	0.026	0.092
cutoff 0.37	7.41 x10^{-3}	0.022	9.26 x10^{-3}

Table 3.5: F-score metric. Cutoff between 0.01 and 0.37 with a stepsize of 0.09 and degree of dynamics (ds) between 1 and 5 with a stepsize of 2.

In Tables 3.3 and 3.4 cutoff 0.01 and ds 3 has the best performance, however, in Table 3.5 the best performance can be seen at cutoff 0.19 and ds at 3. As recall and AUC metrics were more important in this study, F-score only had a little influence on the final values. After comparison of achieved results, the main experiment's cutoff and degree of dynamics values were set to 0.05 and 3.5 respectively. The actual details why recall and AUC values were more important than the others is discussed in the main experiment section.

3.3.3 Main experiment

The following steps were required to perform the experiment:

- Select one of the 12 linked series (as shown in Table 3.2).
- For each crime in the series, send it to Series Finder as a seed.
- Randomly sample 500 crimes from the dataset.
- Crime or crimes are added from current series to the data sample.
- Any existing doubles are removed and the data sample size is re-checked to make sure that it is 500 samples.
- Experiment is performed 10 times with each crime as a seed.

- Previous steps repeated with other crimes from linked series as a seed.

Experiments on the Series Finder, Spatial-Temporal, and Random models were executed following the same procedure. The results were gathered and presented in the results section.

Chapter 4

Results

In this chapter, the outcome of the experiments with regard to the metrics and the statistical tests is presented.

4.1 Metrics results

In the Table 4.3 we can see how Series Finder, Spatial-Temporal, and Random models performed in every run with different metrics.

In Table 4.1 we can see the average performance of 10 runs and standard deviation for Series Finder.

Series Finder	**Average**	**Standard deviation**
Precision	0.025	8.40 x 10^{-3}
Recall	0.096	0.028
F-score	0.038	0.012
AUC	0.543	0.014
ErrorRate	0.013	2.28 x 10^{-4}
Accuracy	0.987	2.28 x 10^{-4}

Table 4.1: Series Finder algorithm's average performance and standard deviation

In Table 4.2 we can see the average performance of 10 runs and standard deviation for Spatial-Temporal.

Spatial-Temporal	**Average**	**Standard deviation**
Precision	0.136	2.93 x 10^{-3}
Recall	0.789	0.0103
F-score	0.225	4.53 x 10^{-3}
AUC	0.886	5.18 x 10^{-3}
ErrorRate	0.017	1.19 x 10^{-4}
Accuracy	0.982	1.19 x 10^{-4}

Table 4.2: Spatial-Temporal model's average performance and standard deviation

Series Finder	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
Precision	0.022	0.039	0.016	0.019	0.038	0.018	0.027	0.017	0.028	0.022
Recall	0.095	0.013	0.065	0.080	0.014	0.070	0.010	0.072	0.010	0.080
F-score	0.037	0.058	0.025	0.030	0.058	0.027	0.041	0.027	0.042	0.033
AUC	0.543	0.562	0.527	0.535	0.570	0.530	0.548	0.531	0.547	0.535
ErrorRate	0.0135	0.0131	0.0138	0.0136	0.0131	0.0135	0.0136	0.0136	0.0134	0.0135
Accuracy	0.987	0.987	0.986	0.986	0.987	0.986	0.987	0.983	0.987	0.986
Spatial-Temporal	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
Precision	0.133	0.136	0.134	0.141	0.139	0.138	0.139	0.133	0.133	0.134
Recall	0.778	0.785	0.787	0.805	0.803	0.793	0.797	0.778	0.778	0.782
F-score	0.220	0.224	0.223	0.233	0.230	0.227	0.229	0.220	0.220	0.222
AUC	0.880	0.884	0.885	0.894	0.893	0.888	0.890	0.880	0.880	0.882
ErrorRate	0.0180	0.0179	0.0179	0.0177	0.0178	0.0178	0.0178	0.0180	0.0180	0.0180
Accuracy	0.980	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982	0.982
Random	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
Precision	0.0057	0.0034	0.0023	0.0023	0.0011	0.0034	0.0085	0.0046	0.0034	0.0023
Recall	0.0211	0.0115	0.0153	0.0153	0.0115	0.0268	0.0498	0.0230	0.0115	0.0153
F-score	0.0089	0.00531	0.0038	0.0038	0.0020	0.0059	0.0013	0.0073	0.0053	0.00386
AUC	0.501	0.496	0.498	0.498	0.496	0.503	0.515	0.501	0.496	0.498
ErrorRate	0.0231	0.0232	0.0232	0.0233	0.0233	0.0232	0.0230	0.0232	0.0232	0.0233
Accuracy	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977	0.977

Table 4.3: Series Finder, Spatial-Temporal and Random models' results for every experimental run. Metrics gathered: precision, recall, F-score, AUC, error rate, accuracy.

In Table 4.4 we can see the average performance of 10 runs and standard deviation for Random model.

Random	**Average**	**Standard deviation**
Precision	3.68 x 10^{-3}	2.01 x 10^{-3}
Recall	0.0201	0.117
F-score	5.99 x 10^{-3}	3.25 x 10^{-3}
AUC	0.500	5.86 x 10^{-3}
ErrorRate	0.023	7.62 x 10^{-5}
Accuracy	0.977	7.62 x 10^{-5}

Table 4.4: Random model's average performance and standard deviation

Figure 4.1 shows box plot representations of the six metrics for all three models as a way of providing an overview of the classification performance of the three candidate models. Note that the y-axis for the accuracy metric does not start at 0 but is rather "zoomed" to show more details. This applies for the box plot that shows AUC metric as well, which starts at 0.5 since lower AUC values do not make sense as such a model's decision-logic just could be inverted in order to provide higher AUC scores.

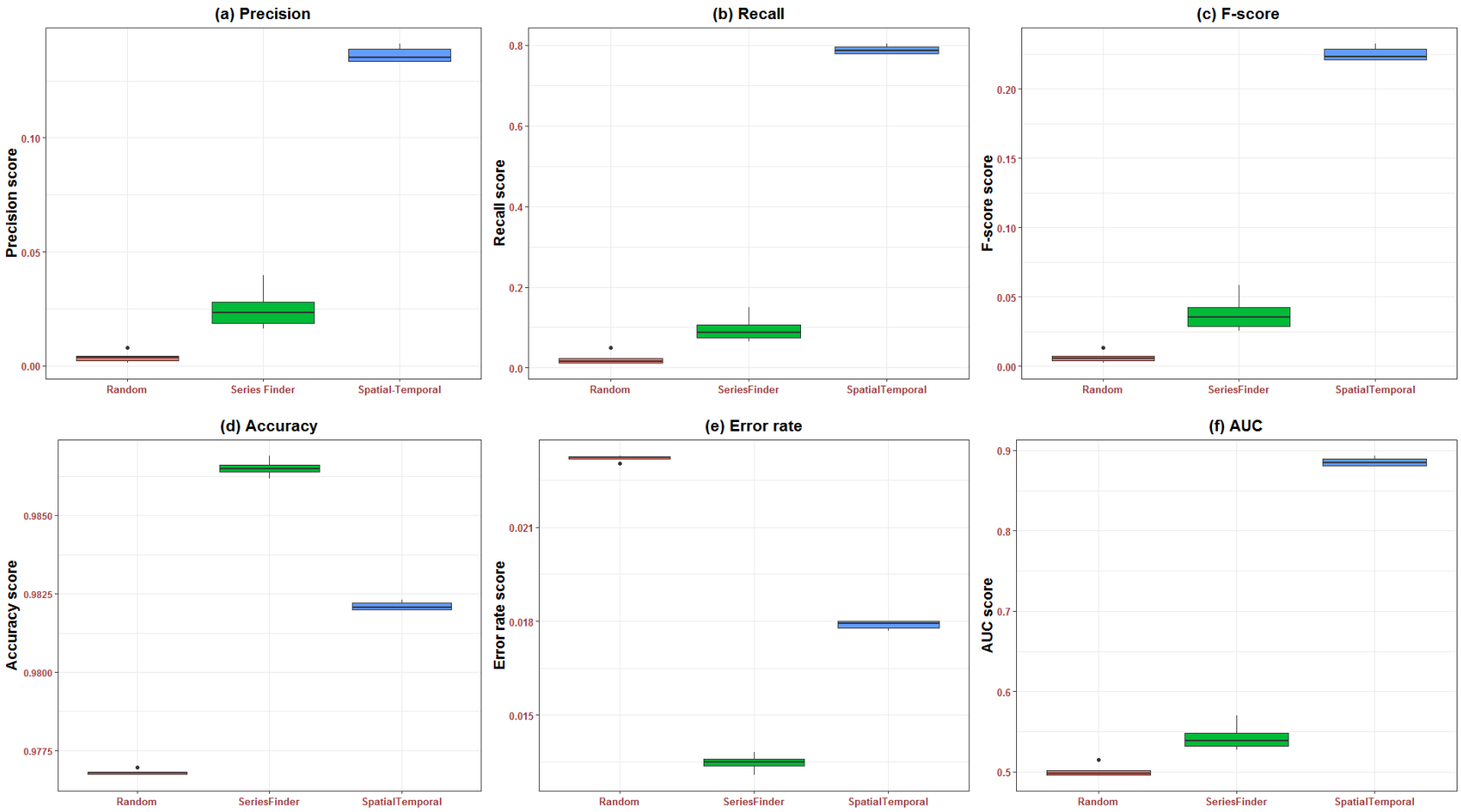

Figure 4.1: One boxplot for each of the following six metrics are shown: (a) precision, (b) recall, (c) F-score, (d) accuracy, (e) error rate, and (f) AUC. In each boxplot, the respective metric scores are shown for the three candidate algorithms evaluated, i.e. Random, Series Finder, and Spatial-Temporal. Values extending 1.5 times the inter-quartile range are regarded as outliers and are shown as black dots in the plot.

4.2 Statistical tests results

After gathering all the metrics' results, we are able to apply statistical tests. Kruskal-Wallis test was chosen because data was not normally distributed and models could not be assumed to have equal variance, which means that the popular ANOVA test was not applicable. In addition, we have measured more than two models, so other tests were not applicable as well.

In a Table 4.5 results from the Kruskal-Wallis test are presented.

Kruskal-Wallis	**statistic value**	**p-value**
Precision	25.887	2.391 x 10^{-6}
Recall	25.881	2.398 x 10^{-6}
F-score	25.858	2.426 x 10^{-6}
AUC	25.840	2.447 x 10^{-6}
ErrorRate	25.869	2.412 x 10^{-6}
Accuracy	25.869	2.412 x 10^{-6}

Table 4.5: Kruskal-Wallis statistical test

In order to understand if the difference exists between models, the $p - value$ has to be lower than the chosen significance level, i.e. 0.05. From the Kruskal-Wallis statistical test, we can see that there is a statistical difference between Series Finder, Spatial-Temporal, and Random models. Therefore, we can apply Nemenyi post-hoc test to find out the significance of the differences. The boundary between significant and insignificant is set to the default value – 0.05. If the $p - value$ is lower than 0.05 it means the difference is significant, otherwise, it is not.

In Table 4.6 we can see the Nemenyi post-hoc results which show how significantly different three models are according to precision metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	0.065
Spatial-Temporal vs Random	2.3 x 10^{-5}

Table 4.6: Nemenyi post-hoc: Precision

According to Table 4.6, it can be seen that the differences in precision between Series Finder vs. Spatial-Temporal and Series Finder vs. Random are not significant. However, the difference between Spatial-Temporal vs. Random is considered to be significant.

In Table 4.7 we can see the Nemenyi post-hoc results which show how significantly different three models are according to recall metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	0.065
Spatial-Temporal vs Random	2.3×10^{-5}

Table 4.7: Nemenyi post-hoc: Recall

According to Table 4.7, it can be seen that the differences in recall between Series Finder vs. Spatial-Temporal and Series Finder vs. Random are not significant. However, the difference between Spatial-Temporal vs. Random is considered to be significant.

In Table 4.8, we can see the Nemenyi post-hoc results which show how significantly different three models are according to the F-score metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	0.065
Spatial-Temporal vs Random	2.3×10^{-5}

Table 4.8: Nemenyi post-hoc: F-score

According to Table 4.8, it can be seen that the differences in F-score between Series Finder vs. Spatial-Temporal and Series Finder vs. Random are not significant. However, the difference between Spatial-Temporal vs. Random is considered to be significant.

In the Table 4.9 we can see the Nemenyi post-hoc results which show how significantly different our three models are according to the AUC metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	0.065
Spatial-Temporal vs Random	2.3×10^{-5}

Table 4.9: Nemenyi post-hoc: AUC

According to Table 4.9, it can be seen that the differences in AUC between Series Finder vs. Spatial-Temporal and Series Finder vs. Random are not significant. However, difference between Spatial-Temporal vs. Random is considered

to be significant.

In Table 4.10 we can see the Nemenyi post-hoc results which show how significantly different our three models are according to the error rate metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	2.3 x 10^{-5}
Spatial-Temporal vs Random	0.065

Table 4.10: Nemenyi post-hoc: Error rate

According to Table 4.10, it can be seen that the differences in error rate between Series Finder vs. Spatial-Temporal and Spatial-Temporal vs. Random are not significant. However, difference between Series Finder vs. Random is considered to be significant.

In Table 4.11 we can see the Nemenyi post-hoc results which show how significantly different our three models are according to the accuracy metric.

Models	**p-value**
Series Finder vs Spatial-Temporal	0.065
Series Finder vs Random	2.3 x 10^{-5}
Spatial-Temporal vs Random	0.065

Table 4.11: Nemenyi post-hoc: Accuracy

According to Table 4.11, it can be seen that the differences in accuracy between Series Finder vs. Spatial-Temporal and Spatial-Temporal vs. Random are not significant. However, difference between Series Finder vs. Random is considered to be significant.

Normally if Cohen's d value is higher than 0.8, the difference is considered to be large [32].

In Table 4.12 it can be seen the Cohen's d effect size between Series Finder and Spatial-Temporal models.

Series Finder vs. Spatial-Temporal	**Cohen's d**
Precision	17.63
Recall	32.67
F-scorel	20.46
AUC	32.19
Error rate	24.41
Accuracy	24.41

Table 4.12: Series Finder vs. Spatial-Temporal Cohen's d effect size

In Table 4.13 it can be seen the Cohen's d effect size between Series Finder and Random models.

Series Finder vs. Random	**Cohen's d**
Precision	3.53
Recall	3.53
F-scorel	3.65
AUC	3.95
Error rate	57.42
Accuracy	57.42

Table 4.13: Series Finder vs. Random model Cohen's d effect size

In Table 4.14 it can be seen the Cohen's d effect size between Spatial-Temporal and Random models.

Spatial-Temporal vs. Random	**Cohen's d**
Precision	52.71
Recall	69.76
F-scorel	55.52
AUC	69.72
Error rate	53.21
Accuracy	53.21

Table 4.14: Spatial-Temporal vs. Random model Cohen's d effect size

Chapter 5

Analysis and discussion

In this section research questions, metrics and results are analyzed. Furthermore, a discussion is presented with reasoning about the obtained results.

5.1 Research question analysis

RQ1. How can Series Finder be implemented in a Swedish context?

After revising the Swedish police dataset and the Series Finder implementation, we noticed that the dataset does not fit it exactly. Series Finder implementation had taken into account how similar the following measurements were: where crime happened, how a crime location was entered, through where the location was entered, when the crime happened, which day of the week the crime happened, type of premises where the crime occurred, was the crime location ransacked, were residents present when the crime occurred, how the suspect looked, how the victim looked and possible time window when the crime happened.

After dataset feature mapping, we noticed that we were unable to use the following parts: how the suspect looked, how the victim looked and the possible time window. Therefore, these parts were removed, so that they did not affect Series Finder main computations. Another modification which was done that implementation required only one crime as a seed, instead of two crimes. This was done because most of our linked series consisted of two crimes. If we had provided two crimes as a seed, we would not be able to verify that the implementation works and finds other belonging crimes to the linked series.

RQ2. How accurate is the modified implementation of Series Finder compared to a state-of-the-art algorithm?

To be able to answer this research question, an experiment was performed. We had the Spatial-Temporal model as a state-of-the-art and the Random model as a baseline comparative. The Random model performed worst in this experiment. The state-of-the-art performed best in this experiment. Series Finder performed in between Spatial-Temporal and Random models. After analyzing metrics and tests results, it could be seen that Series Finder performance was low. Kruskal-Wallis test revealed that the difference exists between all three models. Nemenyi

post-hoc test showed that between Series Finder vs. Spatial-Temporal there was no significant difference on every metric. Series Finder vs. Random had significant difference only on error rate and accuracy metrics. Furthermore, not every metric represented the actual performance as the experiment's nature had limited precision, error rate, and accuracy metrics. The details about the experiment's nature are discussed in section 5.5.

5.2 Performance of the Series Finder model

From the results section, it could be seen that the Series Finder did not perform very well. The first reason for this might be the large difference between the original dataset used for the Series Finder and the Swedish police dataset. Therefore, the modifications which were done to the Series Finder to fit the Swedish police data might have had some impact on overall performance. As after modifications, some of the features were removed or disabled from the Series Finder.

Another possible impact was that originally, the Series Finder had an input of at least two crimes, therefore, calculating more precise crime pattern, instead of only one crime in our case. Finally, we could not know if false positive indeed was performed by the same offender, but thus not solved by the police yet.

5.3 Performance of the Spatial-Temporal model

In the result section, it could be seen that Spatial-Temporal performance was highest compared to other models. As this model performed very well on Swedish police dataset, this means that the linked series were close in time and space or due to a bias in which burglaries the Swedish police manage to solve. This model had a great dependency on the dataset. Therefore, there are no guarantees that it could show similar performance on a different dataset.

Another relevant thing to consider is how many crimes to search for, as this may affect the performance heavily as well. To sum up, this model showed potential based on the values obtained from experiments with the Swedish police dataset.

5.4 Performance of the Random model

In the result section, it could be seen that Random model performed worst. The following could be motivated that in order to find related crimes, proper computations were needed, pure guessing was not enough. This model provided valuable information into future, as any other model can be compared to it to find out if it is worth further investigation.

5.5 Why not all metrics represent model's performance

Usually, every metric brings in the decent amount of information about how an algorithm or model performed. However, this was not the case for this experiment. The proposed experiment limited the actual number of true negatives. All the models were allowed to search for the maximum of ten crimes. Therefore, the true negative rate could not go lower than approximately 2%–2.5% from the total amount. I.e., if we have 500 crimes and allow to search for ten additional crimes, automatically classifying $500 - seedsize - (11 - seedsize)$ crimes as true negatives, we get roughly 2%. This way we can not see the actual performance of error rate, and accuracy, as these metrics are affected by true negatives. Precision is affected by how many crimes algorithms have to search for. In addition, it is impossible to know if some of the false positives are not true positives as police have not solved them.

Chapter 6

Conclusions and future work

As discussed previously, today's society is affected by criminals to a great extent. Therefore, something has to be done. One option to fight crime is to use methods from computer science. In this thesis, computational power has been employed to find out whether it can facilitate police forces. After investigating what has been done already in this field, we have identified a gap which could be explored with an experimental approach. The Series Finder algorithm has not been applied to a Swedish context before. Therefore, this thesis covered the proposed gap.

Series Finder has been applied to Swedish context. Experimental approach revealed that Series Finder performed better than the Random model, but Series Finder performed worse than the state-of-the-art model, i.e. Spatial-Temporal.

As future the Swedish police will add suspect and victim parts in the data, therefore, performance could be different with additional information about the crime. Regarding the timeframe, it could be added when available.

From the gathered results, it can be seen that Series Finder performance is not promising at all. However, there is still some potential left. To gain performance it has to be re-programmed from scratch to fit the Swedish police data exactly. Swedish police data is feature-rich, and its potential is still unused, as in this thesis we have only used a bit more than 1/4 of the available features. Therefore, one objective can be to investigate how Series Finder can use as many features as possible, thus, providing much more accurate insight into modus operandi of every criminal or groups of criminals.

Bibliography

[1] A. Borg, M. Boldt, N. Lavesson, U. Melander, and V. Boeva, "Detecting serial residential burglaries using clustering", *Expert Systems with Applications*, vol. 41, pp. 5252–5266, sep 2014.

[2] S. Sathyadevan, M. Devan, and S. Surya Gangadharan, "Crime analysis and prediction using data mining", pp. 406–412, aug 2014.

[3] T. Wang, C. Rudin, D. Wagner, and R. Sevieri, "Learning to detect patterns of crime", pp. 515–530, Springer, 2013.

[4] S. V. Nath, "Crime Pattern Detection Using Data Mining", pp. 41–44, dec 2006.

[5] M. Munasinghe, H. Perera, S. Udeshini, and R. Weerasinghe, "Machine Learning based criminal short listing using Modus Operandi features", pp. 69–76, aug 2015.

[6] H. Chen, *ACM SIGKDD Workshop on Intelligence and Security Informatics*. New York, NY: ACM, 2010.

[7] J. Demšar, "Statistical comparisons of classifiers over multiple data sets", *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[8] T. Wang, C. Rudin, D. Wagner, and R. Sevieri, "Detecting Patterns of Crime with Series Finder", 2013.

[9] X. Zhou and G. Yu, "Finding criminal suspects by improving the accuracy of similarity measurement", pp. 1145–1149, may 2012.

[10] B. Chandra, M. Gupta, and M. Gupta, "A multivariate time series clustering approach for crime trends prediction", pp. 892–896, oct 2008.

[11] A. Babakura, M. Sulaiman, and M. Yusuf, "Improved method of classification algorithms for crime prediction", pp. 250–255, aug 2014.

[12] F. Ozgul, Z. Erdem, and C. Bowerman, "Prediction of past unsolved terrorist attacks", pp. 37–42, jun 2009.

[13] K. Dahbur and T. Muscarello, "Classification system for serial criminal patterns", *Artificial Intelligence and Law*, vol. 11, no. 4, pp. 251–269, 2003.

[14] D. E. Brown and S. Hagen, "Data association methods with applications to law enforcement", *Decision Support Systems*, vol. 34, no. 4, pp. 369–378, 2003.

[15] D. E. Brown, "The Regional Crime Analysis Program (ReCAP): a framework for mining data to catch criminals", vol. 3, pp. 2848–2853, oct 1998.

[16] J. H. Wang and C. L. Lin, "An Association Model Based on Modus Operandi Mining for Implicit Crime Link Construction", pp. 548–550, jul 2011.

[17] W. Chunyu, W. Xuehua, and Z. Xujuan, "Research on the improved frequent predicate algorithm in the data mining of criminal cases", pp. 1531–1535, jun 2008.

[18] R. K. Boettger and C. Lam, "An overview of experimental and quasi-experimental research in technical communication journals (1992-2011)", *IEEE Transactions on Professional Communication*, vol. 56, no. 4, pp. 272–293, 2013.

[19] "Modus operandi", [Online]. Available: http://www.dictionary.com/browse/modus-operandi?o=O. [Accessed: 15-Mar-2016].

[20] M. Tonkin, J. Woodhams, R. Bull, J. W. Bond, and E. J. Palmer, "Linking Different Types of Crime Using Geographical and Temporal Proximity", *Criminal Justice and Behavior*, vol. 38, pp. 1069–1088, nov 2011.

[21] C. Bruce and R. B. Santos, "Crime Pattern Definitions for Tactical Analysis", pp. 1–5, 2011.

[22] P. A. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge; New York: Cambridge University Press, 2012.

[23] I. H. Witten, E. Frank, and M. A. Hall, *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann series in data management systems, Burlington; MA: Morgan Kaufmann, 3rd ed., 2011.

[24] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks", *Information Processing & Management*, vol. 45, pp. 427–437, jul 2009.

[25] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", pp. 37–63, 2011.

[26] Z. Wang and Z. Xie, "Infrared face recognition based on local binary patterns and Kruskal-Wallis test", pp. 185–188, jun 2014.

[27] J. L. D. Rosa, A. E. A. Magpantay, A. C. Gonzaga, and G. A. Solano, "Cluster center genes as candidate biomarkers for the classification of Leukemia", in *The 5th International Conference on Information, Intelligence, Systems and Applications, IISA 2014*, pp. 124–129, jul 2014.

[28] M. Zhang and Y. Zhou, "Approximate Calculation about Standard Normal Distribution with Genetic Programming", in *2010 Third International Conference on Information and Computing (ICIC)*, vol. 3, pp. 17–20, jun 2010.

[29] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets", *Expert Systems with Applications*, vol. 39, pp. 3446–3453, feb 2012.

[30] T. Baguley, "Standardized or simple effect size: What should be reported?", *ResearchGate*, vol. 100, pp. 17–603, dec 2008.

[31] J. Leppink, P. O'Sullivan, and K. Winston, "Effect size – large, medium, and small", *Perspectives on Medical Education*, vol. 5, pp. 347–349, dec 2016.

[32] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Hillsdale, N.J: Routledge, 2nd ed., jul 1988.

[33] D. Mayorga, M. A. Melgarejo, and N. Obregon, "A Fuzzy Clustering based method for the spatiotemporal analysis of criminal patterns", in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 738–744, jul 2016.

[34] A. K. M. M. Islam, T. Nakai, and H. Onodera, "Statistical analysis and modeling of Random Telegraph Noise based on gate delay variation measurement", in *2016 International Conference on Microelectronic Test Structures (ICMTS)*, pp. 82–87, mar 2016.