Effective mining of crime patterns from growing volumes of data using improved FP-growth algorithm

Matto, George

NM-AIST

http://dspace.nm-aist.ac.tz/handle/123456789/303

Downloaded from Nelson Mandela-AIST's institutional repository
EFFECTIVE MINING OF CRIME PATTERNS FROM GROWING VOLUMES OF DATA USING IMPROVED FP-GROWTH ALGORITHM

George Matto

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

February, 2019
ABSTRACT

The spate of crimes in Tanzania, as in many other countries, has been on the increase in the last few years. The successes recorded by criminals have been attributed by improper mechanisms for crime detection, prevention and control. Proactive measures are needed to preempt further crimes. Frequent pattern mining stand to aid in finding emerging patterns, series, and trends in the crime data. This will eventually help Tanzania Police Force and other law enforcement agencies to understand crime trends and predict or forecast future occurrences and thus improve preventive measures against crimes. FP-Growth is the most effective and most widely used algorithm for frequent pattern mining and association rules generation. Unfortunately, studies have shown two main weaknesses associated with this algorithm when used in the growing volumes of crime data. Inability of the algorithm to scale-up well with the growing volumes data is the first weakness. Second weakness of FP-Growth is associated with the nature of crime data. Crime data consists of items (different crimes) that vary greatly in terms of frequencies of occurrence. Some of crimes (e.g. robbery) happen so frequently and thus frequently appear in the dataset while other crimes (e.g. killing of people with albinism, in the case of Tanzania) happens seasonally and therefore rarely found in the dataset. Classical FP-Growth algorithm extract frequent patterns by using single user-defined minimum support. This is the main source of the algorithm’s challenge, especially when used in crime datasets. To tackle the challenges, this study have proposed a multiple minimum support FP-Growth algorithm that scans the dataset and automatically assigns minimum support values to each crime item basing on how frequently it has appeared in the dataset. The proposed solution is based on the Shannon Entropy in which an algorithm for obtaining multiple item support values was developed. In connection, the study developed a working prototype that was based on the proposed approach to help in recording crime data and mine crime patterns from data. The proposed approach was evaluated against classical FP-Growth as well as CFPGrowth algorithm on the varying sizes crime data. Evaluation results showed that the proposed approach in this research is more efficient in mining frequent crime patterns, and more effective in terms of run-time and memory use. Basing on the results found, the study recommends for further experimentation of the proposed approach on streaming data and on distributed environments, among other recommendations as stipulated in the dissertation.
DECLARATION

I, GEORGE MATTO do hereby declare to the Senate of Nelson Mandela African Institute of Science and Technology that this dissertation is my own original work and that it has neither been submitted nor being concurrently submitted for degree award in any other institution.

George Matto

Name and Signature of Candidate  Date

The above declaration is confirmed

Dr. Joseph Mwangoka

Name and Signature of Supervisor  Date
COPYRIGHT

This dissertation is copyright material protected under the Berne Convention, the Copyright Act of 1999 and other international and national enactments, in that behalf, on intellectual property. It must not be reproduced by any means, in full or in part, except for short extracts in fair dealing; for researcher private study, critical scholarly review or discourse with an acknowledgement, without the written permission of the office of Deputy Vice Chancellor for Academics, Research and Innovations, on behalf of both the author and the Nelson Mandela African Institution of Science and Technology.
CERTIFICATION

The undersigned certify that he has read and hereby recommend for examination of a dissertation titled EFFECTIVE MINING OF CRIME PATTERNS FROM GROWING VOLUMES OF DATA USING IMPROVED FP-GROWTH ALGORITHM in fulfilment of the requirements for the degree of PhD in Information and Communication Science and Engineering at the Nelson Mandela African Institution of Science and Technology.

Dr. Joseph Mwangoka

Name and Signature of Supervisor                  Date
ACKNOWLEDGEMENTS

I am very thankful to Almighty God for not only granting me this rare opportunity to join a PhD programme, but also giving me physical, mental and spiritual health that allowed me to accomplish the journey.

I am also very grateful to my first supervisor, Dr. Joseph Mwangoka, for the tireless supervision, coaching and advice he has given me since very early stages of the proposal to this final reporting. He was always available for consultation even while away or during very tight schedules. I feel so blessed and privileged to have Dr. Mwangoka as not only my supervisor but also a mentor and a role model.

In a very special way I wish to thank Dr. Khamis Kalegele from the Tanzania Commission for Science and Technology (COSTECH) and former NM-AIST Lecturer and Dr. Shubi Kaijage (Dean, CoCSE) for their technical inputs that shaped the study. I am also grateful to Eng. Dr. Yonah Zaipuna for the useful advices and encouragements. My thanks go also to the NM-AIST (Schools of CoCSE and BuSH) academic bench for the very insightful knowledge that I acquired during coursework part of my study.

My PhD study was sponsored by the Tanzania Communications Regulatory Authority (TCRA) through TCRA ICT Scholarships. I am very grateful to the TCRA for making sure my studies went as smooth as possible in terms of financial support. I also would like to thank my employer, Moshi Co-operative University, for releasing me to pursue my PhD studies.

Very special thanks go to the Tanzania Police Force, Head Office for providing me with some of data that were needed in this study as well as taking part in the end user evaluation of the developed prototype. Afande Saida and Inspector Johnson, thank you very much on behalf of others. In connection, I am very grateful to the Information Systems Development experts who tested and technically evaluated our developed system. These experts provided useful technical advices. On behalf of others, I sincerely thank Mr. Pascal Msoka for the useful technical inputs.

And finally, last but by no means least, my heartfelt thanks go to my family. Your advices, prayers and moral support were always igniting in me a fresh spirit to press on.
DEDICATION

To my family.
TABLE OF CONTENTS

ABSTRACT ................................................................................................................................. i

DECLARATION .......................................................................................................................... i

COPYRIGHT ........................................................................................................................... iii

CERTIFICATION ....................................................................................................................... iv

ACKNOWLEDGEMENT ............................................................................................................. v

DEDICATION ............................................................................................................................. vi

TABLE OF CONTENTS .............................................................................................................. vii

LIST OF TABLES ....................................................................................................................... xi

LIST OF FIGURES ..................................................................................................................... xii

LIST OF ABBREVIATIONS AND SYMBOLS ............................................................................ xv

CHAPTER ONE: INTRODUCTION .................................................................................................. 1

1.1 Background Information ...................................................................................................... 1

1.2 Research Problem and Justification of the Study ................................................................. 6

1.3 Objectives ............................................................................................................................. 6

1.3.1 General Objective ....................................................................................................... 6

1.3.2 Specific Objectives .................................................................................................... 6

1.4 Research Questions ............................................................................................................. 7

1.5 Significance of the Research ............................................................................................. 7

1.6 Organization of the Dissertation ....................................................................................... 8

CHAPTER TWO: DETECTING CRIME PATTERNS FROM DATASETS IN TANZANIA USING FREQUENT PATTERN MINING .......................................................................................... 11
2.1 Introduction ............................................................................................................................. 11
2.2 Literature Review .................................................................................................................... 14
2.3 Research Methodology ........................................................................................................... 16
  2.3.1 Source of Data .............................................................................................................. 16
  2.3.2 Workflow of the Crimes Mining Process ................................................................ 16
  2.3.3 Association Rules Generation ................................................................................... 20
2.4 Results ..................................................................................................................................... 20
  2.4.1 Crime Occurrence Frequencies ................................................................................ 20
  2.4.2 Crime Occurrence by Regions ................................................................................. 24
  2.4.3 Association Rules Generated ................................................................................... 26
2.5 Effects of Increasing Dataset Size on Execution Time ........................................................... 27
2.5 Chapter Conclusion ................................................................................................................. 29

CHAPTER THREE: MINING FREQUENT PATTERNS OF CRIME USING MULTIPLE
MINIMUM SUPPORTS FP-GROWTH BASED ON SHANNON ENTROPY ....................... 31

3.1 Introduction ............................................................................................................................. 31
3.2 Related Literature .................................................................................................................... 33
  3.2.1 Association Rules Mining ........................................................................................ 33
  3.2.2 FP-Growth Algorithm .............................................................................................. 34
  3.2.3 Studies to improve FP-Growth Algorithm ............................................................... 38
  3.2.4 Approaches that use Multiple Minimum Support .................................................... 40
3.3 The Proposed Approach .......................................................................................................... 41
3.4 Experimental Results .............................................................................................................. 44
3.5 Chapter Conclusion ................................................................................................................. 47
CHAPTER FOUR: FRAMEWORK FOR EXTRACTION OF FREQUENT PATTERNS OF CRIME FROM MULTIPLE SOURCES OF DATA USING ENTROPY BASED FP-GROWTH ALGORITHM ................................................................. 48

4.1 Introduction .................................................................................................................................. 48
4.2 Related Works .............................................................................................................................. 49
4.3 The Proposed Framework ............................................................................................................ 50
  4.3.1 Data Sources ...................................................................................................................... 51
  4.3.2 Data Cleaning and Pre-processing .................................................................................. 52
  4.3.3 Pattern Generation ........................................................................................................... 54
  4.3.4 Pattern Evaluation and Visualization .............................................................................. 56
4.4 Framework Validation .................................................................................................................. 77
4.5 Chapter Conclusion ..................................................................................................................... 78

CHAPTER FIVE: CRaPES: AN FP-GROWTH BASED PROTOTYPE FOR CRIME REPORTING AND PATTERNS EXTRACTION IN TANZANIA ................................................................. 49

5.1 Introduction .................................................................................................................................. 49
5.2 Methodology ................................................................................................................................ 60
5.3 CRaPES Implementation ............................................................................................................. 61
  5.3.1 Crime Reporting Module ............................................................................................... 53
  5.3.2 Pattern Extraction Module ............................................................................................. 65
  5.3.3 Exporting Crime Dataset and MIS File .......................................................................... 67
5.4 Results and Description of the Patterns Extracted ...................................................................... 70
5.5 CRaPES Evaluation ...................................................................................................................... 71
  5.5.1 Technical Evaluation ........................................................................................................ 71
  5.5.2 End-User Evaluation ......................................................................................................... 72
5.6 Chapter Conclusion ..................................................................................................................... 73
CHAPTER SIX: GENERAL DISCUSSION, CONCLUSION AND RECOMMENDATIONS

6.1 General Discussion ................................................................................................................. 79

   6.1.1 Frequent Pattern Mining for Crime Detection ................................................................. 74
   6.1.2 FP-Growth Improvement Method ..................................................................................... 75
   6.1.3 Generic Framework for Mining Frequent Patterns of Crime ........................................ 76
   6.1.4 Crime Pattern Mining Prototype ..................................................................................... 77

6.2 General Conclusion ................................................................................................................. 77

6.3 General Recommendations ..................................................................................................... 78

6.4 Future Works ........................................................................................................................ 79

   6.4.1 More Datasets and More Data Sources ........................................................................... 79
   6.4.2 Live, Streaming and Big Data Processing ........................................................................ 79
   6.4.3 Mobile and Web-based Applications for Crime Patterns Mining ................................... 80

REFERENCES .................................................................................................................................... 81

APPENDICES ................................................................................................................................... 89

RESEARCH OUTPUTS ..................................................................................................................... 104
LIST OF TABLES

Table 1: Horizontal Data Format .................................................. 3
Table 2: Vertical Data Format ....................................................... 4
Table 3: Mined crimes from MAJIRA newspaper (April/May, 2016) ................. 22
Table 4: Mined crimes from MTANZANIA newspaper (April/May, 2016) ............ 22
Table 5: Mined crimes from MWANANCHI newspaper (April/May, 2016) .......... 23
Table 6: Mined crimes from NIPASHE newspaper (April/May, 2016) ............... 23
Table 7: Different dataset sizes used ............................................. 28
Table 8: Dataset in the Transaction Database (DB) ................................ 37
Table 9: Mined Patterns (with $\xi = 2$) ........................................... 39
Table 10: The proposed algorithm for specifying MIS values ......................... 43
Table 11: Mined Patterns (with the proposed MIS approach, $LS = 2$) ............ 44
Table 12: Memory Use in MB (Proposed Approach vs. FP-Growth) ................. 45
LIST OF FIGURES

Figure 1: Organization of the Dissertation .................................................. 9
Figure 2: Trend of crimes reported to police stations, from 2011 to 2015 ............ 12
Figure 3: Trend of crime fearing rate in Tanzania, from 2010 to 2014 ............... 13
Figure 4: Different tokens replaced by mauaji .............................................. 18
Figure 5: Workflow of the proposed solution ................................................. 18
Figure 6: Sample data prepared for rules generation .................................... 19
Figure 7: Operators involved in the association rules generation process ......... 20
Figure 8: Number of newspapers’ articles that reported the crimes ................. 24
Figure 9: Map of Tanzania showing crimes distribution in April/May, 2016 .... 25
Figure 10: Graphical visualization of generated association rules ................. 26
Figure 11: Crude dataset ............................................................................. 28
Figure 12: Cleaned and Pre-processed dataset ............................................ 29
Figure 13: Graphical representation of association rules generated from a 100MB dataset ................................................................. 29
Figure 14: Execution time (in seconds) on different datasets ....................... 30
Figure 15: A complete FP-Tree for a crime database (DB) .......................... 38
Figure 16: Execution Time in ms (Proposed Approach vs. FP-Growth) .......... 45
Figure 17: Execution Time in ms (Proposed Approach vs. CFPGrowth) .... 46
Figure 18: Memory Use in MB (Proposed Approach vs. CFPGrowth) .......... 46
Figure 19: The Proposed Framework .......................................................... 51
Figure 20: Flowchart for obtaining MIS values ......................................... 55
LIST OF APPENDICES

Appendix 1  Interview Guide Questions. .................................. 89

Appendix 2  Sample Newspaper Articles Used - Mtanzania Newspaper. .......... 92

Appendix 3  Sample Newspaper Articles Used - Nipashe Newspaper. .......... 93

Appendix 4  Sample Newspaper Articles Used - Mwananchi Newspaper. ........ 94

Appendix 5  Source Codes for MIS Tree Construction. .......................... 95

Appendix 6  Source codes for Calculating MIS values by using Entropy equation. .... 98

Appendix 7  Source codes for CRaPES Manage Crimes. .......................... 99

Appendix 8  Source codes for CRaPES Report Crimes. .......................... 101
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRaPES</td>
<td>Crime Reporting and Pattern Extraction System</td>
</tr>
<tr>
<td>DB</td>
<td>Transaction Database</td>
</tr>
<tr>
<td>FP</td>
<td>Frequent Pattern</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>ID</td>
<td>Identity</td>
</tr>
<tr>
<td>kNN</td>
<td>k-nearest neighbors</td>
</tr>
<tr>
<td>LS</td>
<td>Least Support</td>
</tr>
<tr>
<td>MIS</td>
<td>Multiple item Support</td>
</tr>
<tr>
<td>MLP</td>
<td>Multilayered Perceptron</td>
</tr>
<tr>
<td>NBS</td>
<td>National Bureau of Statistics</td>
</tr>
<tr>
<td>SD</td>
<td>Support Difference</td>
</tr>
<tr>
<td>SOM</td>
<td>Self-Organizing Map</td>
</tr>
<tr>
<td>TID</td>
<td>Transaction ID</td>
</tr>
<tr>
<td>TPF</td>
<td>Tanzania Police Force</td>
</tr>
<tr>
<td>UN</td>
<td>United Nations</td>
</tr>
<tr>
<td>URT</td>
<td>United Republic of Tanzania</td>
</tr>
<tr>
<td>ξ</td>
<td>Minimum Support Threshold</td>
</tr>
</tbody>
</table>
CHAPTER ONE

INTRODUCTION

This first chapter of the dissertation presents the general introduction of the research. Specifically, it describes the background of this study, research problem and justification, objectives of the study, and research questions. It also presents significance of the research as well as organization of this dissertation.

1.1 Background Information

The term crime does not have a simple and universally accepted definition. This study considers crime, as Henry and Lanier defines, an intentional act in violation of the criminal law, committed without defense or excuse, and penalized by the state (Henry and Lanier, 2001). According to this definition, an act is considered crime if declared as such by the relevant law in a particular domain. Crimes are of different forms. This study focused specifically on violent crimes (such as murder and non-negligent manslaughter, sexual offenses, robbery, and aggravated assault) and property crimes (such as burglary, larceny-theft, and motor vehicle theft). Thus, the term crime in this study generally refers to violent and property crimes.

Crimes have got diverse effects not only to the crime victim but also to the community. According to Wasserman and Ellis (2008) crime victims suffer immediate physical or financial problems as well as long-term stress and anxiety. On the community side, criminal behaviors reduce perception of safety and steadily increase the fear of crime that results into social and economic effects. According to United Republic of Tanzania URT (2016) and Weatherburn (2001) there exists a direct relationship between peace and security and socio-economic development; reduced crimes entail more economic opportunities and thus socio-economic development, and vice versa.

All countries around the world strive to ensure safety and security for their citizens by maintaining law and order, and detecting and preventing crime so as to create an enabling environment
for people to engage themselves fully in productive works (URT, 2016). Unfortunately, according to Isafiade and Bagula (2013), most of crime prevention strategies in many developing countries such as Tanzania rely on inadequate tools and practices, and in most cases are based on manual or personal judgments.

As the World is in the era of immense developments in science and technology, criminals in Tanzania, as in many other countries, employ technologies to plan, organize and execute their actions. In this way, in most cases traditional methods for crime detection and prevention used by police are escaped by the criminals. But, as Nwanga et al. (2014) pointed out, most of such criminal activities generate data through electronic platforms like emails, social networks, web logs, phone calls, and telecommunication facilities. Such data leave traces necessary for tracking the criminals. On the other hand, the general public through mass media, social media and other similar platforms reports crimes that happen in the communities. In so doing they generate data that when analyzed can provide useful insights that can be helpful to the law enforcement agencies. Likewise, as all police in the World, the Tanzania Police Force (TPF) keeps several crime related records (such as the types of crimes committed (W-what), locations in which they happened (W-where), time in which they happened (W-when), the causes of the such crimes (W-why), details of the criminal(s) (W-who) involved (e.g. criminal’s name, age, residence, marital status, etc.), and how was it committed (H-how)). Such records are normally referred by the TPF as 5WH of crime. Analysis of the stored 5WH of crime can generate useful insights, patterns and trends that stand to assist TPF and other law enforcement agencies in the country to improve strategies for crime detection and prevention.

Frequent pattern mining is a practical approach that can be employed to extract crime patterns from datasets. A frequent pattern is a pattern (set of items, subsequences, or substructures) that occurs frequently in a dataset (Han and Kamber, 2006). Frequently occurrence means that a specific item or set of items appears in a dataset with frequency no less than specified minimum threshold. For example, if a set of items such as ‘theft’ and ‘body injury’ appear in the dataset three times or more (if three is a minimum support threshold) are said to be frequent itemsets. Finding frequent itemsets is an essential step towards generating association rules and many other interesting relationships between items in the dataset. According to Han et al. (2000) an
association rule is an implication of the form \( X \rightarrow Y \), where, \( X \) is the antecedent, while \( Y \) is the consequent. It provides information of the form; if \( X \) happens then \( Y \) will possibly happen as well. For instance, a simplified form of an association rule can be; *robbery (in the shop) → murder (in the shop)*, which can mean that criminals involved in robbery, in the shop, end up killing the victims.

Numerous techniques and algorithms for extracting frequent itemsets and generating association rules have been proposed. According to Kumbhare and Chobe (2014), Apriori algorithm proposed by Agrawal and Srikant (1994), Eclat algorithm proposed by Zaki et al. (1997), and FP–Growth algorithm proposed by Han et al. (2000) are among the best known and most widely used association rule mining algorithms. Apriori algorithm, which is based on breadth-first search (horizontal data format, Table 1), works by generating a \((k+1)\)-itemset from frequent \( k \)-itemsets, until no more frequent itemsets can be found. The number of candidate itemsets generated limits the Apriori’s efficiency. Eclat uses depth-first search (vertical data format, Table 2) and works by transforming the dataset from horizontal format to vertical, then uses tidset (set of Transaction IDs (TIDs) whose transactions contain the item) intersections to compute the support of a candidate itemset. In this way, unlike Apriori, Eclat avoids generation of subsets that does not exist in the prefix tree (Zaki et al., 1997).

Table 1: Horizontal Data Format

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>I1, I2, I4</td>
</tr>
<tr>
<td>T2</td>
<td>I1, I2</td>
</tr>
<tr>
<td>T3</td>
<td>I3, I4</td>
</tr>
<tr>
<td>T4</td>
<td>I2, I3</td>
</tr>
</tbody>
</table>

Different from Apriori and Eclat algorithms, FP–Growth algorithm does not need to generate candidate itemsets; instead, it uses a divide–and–conquer approach by compressing datasets into a Frequent Pattern Tree (FP–tree) and obtains frequent patterns from the generated FP–tree (Han et al., 2000). FP–Growth works in two steps; first is mining of frequent itemsets, and second is generation of association rules amongst the mined itemsets. Studies have established
Table 2: Vertical Data Format

<table>
<thead>
<tr>
<th>Items</th>
<th>tidsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>{T1, T2}</td>
</tr>
<tr>
<td>I2</td>
<td>{T1, T2, T4}</td>
</tr>
<tr>
<td>I3</td>
<td>{T3, T4}</td>
</tr>
<tr>
<td>I4</td>
<td>{T1, T3}</td>
</tr>
</tbody>
</table>

FP–Growth as the most effective association rule mining algorithm compared to Apriori and Eclat (Isafiade and Bagula, 2013; Kumbhare and Chobe, 2014). FP–Growth algorithm have also been widely employed in extracting crime insights—for discovering trends in criminal behavior and predicts or anticipates criminals’ activities—from raw data.

Studies have shown Tanzania as one of the countries with an increased digital use and thus enormous data generation. For instance, according to Kemp (2018), annual active social media users in Tanzania have grown by +79% (2 million new social media users) in only one year between January 2016 and January 2017. Similarly active mobile social users have grown by +91% between January 2016 and January 2017. By January 2017, mobile subscriptions in the country was 40.10 (out of 56.02) million Tanzanians. In this case, enormous amounts of data are increasingly generated in the country. These data are potential sources for the extraction of useful patterns of crime.

As it has already been pointed out, FP-Growth algorithm is a very potential algorithm for the extraction of crime patterns from crime data. Thus, this algorithm can be employed by the Tanzania Police to discover valuable crime insights from the growing volumes of data. This could, in consequence, support them as well as other law enforcement agencies in the country to enhance crime detection and prevention strategies.

Unfortunately, scholars (Gawwad et al., 2017; Chen et al., 2009; Samra and Maghari, 2015) have shown that when the dataset grows, effectiveness of FP-Growth decreases. This limits effectiveness of the algorithm is the growing volumes of data. It was on this ground that the present study was carried out to propose ways in which FP-Growth algorithm can be improved
for effective mining of frequent patterns of crime from growing volumes of data.

1.2 Research Problem and Justification of the Study

Although FP-Growth has been extensively used in frequent pattern mining as equally as in crime pattern mining (Isafiade and Bagula, 2013; Kumbhare and Chobe, 2014), studies have shown that its effectiveness decreases as the amount dataset increases (Gawwad et al., 2017; Chen et al., 2009; Samra and Maghari, 2015). The main contributing factor for this is related to how the minimum supports of the items are specified in the FP-Growth algorithm. Classical FP-Growth generates frequent itemsets by using a single user-defined minimum support. This approach is not adequate for real life applications, particularly in crime patterns mining. The reasons are twofold. First, the amount of crime data continues to grow, and as they grow the existing approaches require a continual changing and fine-tuning the minimum support as per the growing data. According to Chen et al. (2014) this process is impractical, tedious and time consuming. Second, crime data consists of items (crime types) that differ greatly in terms of frequency of occurrences in the datasets. Some items, such as ‘robbery’, are regularly committed and thus frequently appear in the dataset while other items, such as ‘albino killing’ in the case of Tanzania, happen seasonally \(^1\) and thus rarely found in the dataset. In this case, a single user–defined minimum support is not practical. To accommodate the two items, it may be set too low (to ensure patterns related to ‘albino killing’ are taken on board) or set too high (to limit ‘robbery’–related patterns). However, if it is set too low, huge amount of crime patterns (including uninteresting patterns) may be generated; on the other hand, if it is set too high, many interesting patterns (including those that appear seasonally like ‘albino killing’) may be lost.

Studies have proposed the replacement of a single minimum support threshold in FP-Growth with multiple minimum supports. As a result, algorithms like CFPGrowth that uses percentage based approach to obtain Multiple Item Support (MIS) values, and CFPGrowth++ were proposed. The main idea of CFPGrowth++ is the use of the concept of “Support Difference” (SD) instead of a percentage-based methodology, to specify items’ MIS values. Although the

\(^1\)According to Makoye (2015) the number of albino killing in Tanzania rises during general elections
proposed approaches have shown some improvements, they are still facing the challenge of specifying “good” support value for each item in the dataset. According to Chen et al. (2004), for example, such algorithms require user to identify a minimum support value for each item and constantly tune that support value. This is a costly work as it takes time and efforts to scan the database and then to re-execute the algorithms.

In this case, although FP-Growth has been proven to be useful in frequent pattern mining, it cannot be effectively used to mine frequent crime patterns in growing volumes of crime data (Gawwad et al., 2017; Samra and Maghari, 2015). While this is the situation, research has identified a continued rise of crime fear (from 39% in 2010 to 45% in 2014, as it is described in more details in Chapter Two). Scholars have established a direct relationship between increased crime fear and increased criminal incidents (Jackson, 2009). Therefore, despite of the existing crime prevention methods in the country, reports on the continued increase of crime fear imply a continued increase of criminal incidents. This study has proposed the employment of frequent pattern mining, through best, efficient and effective data mining algorithm, to enhance country’s crime prevention strategies.

1.3 Objectives

1.3.1 General Objective

The general objective of this research was to improve FP-Growth algorithm for effective mining of frequent crime patterns in the growing volumes of data in Tanzania.

1.3.2 Specific Objectives

(i) To explore ways in which frequent pattern mining can be useful in detecting crime patterns from available datasets in the country.

(ii) To propose efficient FP-Growth scaling method for effective mining of frequent patterns of crime.
(iii) To propose a generic framework for mining crime patterns from multiple sources of data.

(iv) To develop pattern mining prototype basing on the proposed FP-Growth scaling method and proposed framework.

1.4 Research Questions

(i) How can frequent pattern mining help in detecting useful crime patterns from available datasets in Tanzania?

(ii) How can FP-Growth algorithm be scaled-up to be effective in extracting frequent patterns of crime?

(iii) How can a generic framework for mining frequent patterns of crime from multiple crime data sources be developed?

(iv) How can a pattern mining prototype that is based on the proposed FP-Growth scaling method be developed?

1.5 Significance of the Research

Crime detection and prevention is the fundamental responsibility of any police in the World. Through this study, the Tanzania Police Force (TPF) and other law enforcement agencies in the country will be assisted in improving their strategies for crime detection and prevention in the following ways. First, the study have shown how data outside police departments can be helpful in providing useful crime insights. This will help the law enforcement agencies to rely not only on their crime data but also on other potential sources. Second, the study have proposed and developed a working prototype that can help police force to improve reporting and managing crime data in an electronic platform. Through the developed prototype, and the pattern mining model, TPF can also be able to effectively analyze their available crime datasets and discover useful insights from the data. That being so, the study stands to contribute to reducing crime rates in the country.
On the other hand, this study is going to make a significant contribution to the community of data scientists and analysts through the FP-Growth scaling method and the crime pattern mining framework proposed. The proposed scaling method allows FP-Growth to assign automatically minimum support values to each of the crime item in the dataset depending on how they have appeared in the dataset. The proposed crime pattern mining framework provides guideline to crime analysts on the experimental processes when extracting crime patterns from both structured and untrusted data. This will consequently contribute to the existing body of knowledge in this area.

1.6 Organization of the Dissertation

As Figure 1 shows, this dissertation is organized in six inter-related chapters. Chapter One presents the general introduction of the research. The chapter includes, background of the study, research problem and justification, objectives of the study, research questions, significance of the study, and organization of the dissertation.

Chapter Two examines ways in which frequent pattern mining can be helpful in detecting useful patterns of crime from available datasets in the country. The chapter describes how frequent crime patterns can be extracted from datasets through a developed pattern mining model. As a result of findings from this chapter one research paper titled Detecting crime patterns from Swahili newspapers using text mining was published in the International Journal of Knowledge Engineering and Data Mining (Matto and Mwangoka, 2017).

Chapter Three presents the proposed method for improving FP-Growth algorithm. Basing on the results that was observed in Chapter Two on the effects of increasing dataset size on FP-Growth’s performance, this Chapter proposed an FP-Growth scaling method that is based on the use of multiple minimum supports approach. Specifically, Shannon Entropy (Information Entropy) was employed to obtain the MIS values of the crime items in the dataset. The proposed FP-Growth scaling method is published on a paper titled Mining Frequent Patterns of Crime using FP-Growth with Multiple Minimum Supports based on Shannon Entropy in the International Journal of Computer Applications (Matto and Mwangoka, 2018).
In Chapter Four, the study proposes a generic framework for the extraction of crime patterns from multiple data sources. The framework is proposed to confront the challenge that were observed in most of the existing similar frameworks. The framework is validated basing on the experimental results that were obtained in Chapter Two and Three which is further validated in Chapter Five.

Chapter Five investigates effectiveness of the FP-Growth scaling method proposed in Chapter Three, and partly validates the patterns mining framework proposed in Chapter Four. The Chapter came up with a working prototype for reporting crimes and extracting patterns stored data. The prototype is named *Crime Reporting and Pattern Extraction System (CRaPES)*. This prototype was evaluated by both users and information systems development experts and found
to be useful for crime patterns mining.

Chapter Six presents the general discussion, general conclusion, general recommendations, and areas for further research, as detailed in the dissertation.
CHAPTER TWO

DETECTING CRIME PATTERNS FROM DATASETS IN TANZANIA USING
FREQUENT PATTERN MINING

This Chapter shows how frequent pattern mining can be helpful in detecting useful crime insights from available datasets outside police departments. The Chapter specifically focuses on the news articles in newspapers. With the use of the developed frequent patterns mining model several crimes reported in the newspapers were extracted. Also, the distribution of the mined crimes per regions of the country were mapped, and with the help of FP-Growth, association rules between the mined crimes were generated.

2.1 Introduction

Frequent pattern mining is the process of extracting sets of items, subsequences or substructures that frequently occur in a dataset. Mining frequent patterns of crime can help police and other law enforcement agencies to improve approaches for crime detection, prevention and control. But, while research shows that more than half of crimes committed in Tanzania are not reported to police, there is a great reliance by such law enforcement agencies on the analysis of datasets of only reported crimes.

Reports from the Tanzania Police Force (TPF) and the National Bureau of Statistics (NBS) show a slight decrease of crimes reported to police stations in recent years. For instance, as Fig. 2 shows, the number of criminal offences reported in 2015 was 519,203, compared with 528,575 that was reported in 2014, a decrease which was equivalent to 1.8%. Similarly, there was a 6% and 1.1% decrease of crimes reported in 2014 and 2013 respectively (URT, 2013, 2014, 2015, 2016). While such reports record a crime decrease, several researches indicate a continued rise of crime fear among Tanzanians (Wambura, 2015a). Crime fear is defined by Hale (1996) as

---

the fear of being a victim of crime. In 2011, for example, 42% of Tanzanians were living with fear of becoming victims of crimes (Wambura, 2015a), 41% in 2012 (Gaddis and Wane, 2013), and over 45% in 2014 (Twaweza, 2014) as described more in Fig. 3.

Reports about reduced number of reported crimes on one hand, and those of increased crime fear on the other hand presents a contradicting fact about the actual crime situation in the country. However, Jackson (2009) argues that there exists a link between fear of crime and likelihood of victimization, and that, high crime fearing rate is a natural response to crime incidents as it is grounded on the reality of crime. This is why Twaweza (2014) pointed out that the increased crime fear in Tanzania is a result of the increased crime incidents. In connection, research shows a low tendency of crime reporting to police in Tanzania. Some of the identified underlying reasons for that are inaccessibility of police stations, unresponsiveness of the police, and police corruption (Aiko, 2015). In 2011 to 2013 for example, 54% of people who were victims of crime did not report the incidents to police (Wambura, 2015b).

![Figure 2: Trend of crimes reported to police stations, from 2011 to 2015](image)

Although police reports show a decreasing rate of reported crimes, the incidents are likely to be on the increase. It is not surprising therefore that many Tanzanians, as reported by OSAC (2016), have taken self-initiatives to protect themselves against crime. Majority of them use
neighborhood watch, the people’s militia (or *Sungusungu*), weapons, security guards, dogs, robust fences/walls, robust grilles, and electric fences or broken bottles as crimes preventive measures (OSAC, 2016).

Vigorous proactive measures are needed to reinforce the prevention of further criminal incidents. As Zaman (2013) pointed out, crime prevention is the primary objective of an efficient police. The advancements in science and technology can play a major role to support this. Technologies can help in analyzing crime datasets to find emerging patterns, series, and trends. This will help police force to understand the current trends in criminal activities and predict or forecast future crime occurrences (Usha and Rameshkumar, 2014).

Unfortunately, Tanzania Police Force as many other law enforcement agencies in developing countries rely mostly on manual or personal judgments and other inadequate tools for inspection, exploration and analysis of crime data. Moreover, as argued by Isafiade and Bagula (2013), the volume of data that can be processed simultaneously within a reasonable time frame is limited thus results into omission of complex and crucial relationships between different crimes attributes. Apart from the challenge of maximally exploring crime datasets, the Tanzania Police Force relies mostly on reported crimes. Capturing and analyzing unreported crimes have remained a challenge.

---

*Sungusungu* are Tanzanian justice organizations established originally by the Sukuma and Nyamwezi ethnic groups in the 1980s to protect cattle from theft and other property (Abrahams, 1987).
Newspapers, social media, and other similar platforms can be very useful sources of crime data that are not necessarily reported to police stations. Employment of frequent pattern mining on such datasets can provide police with very useful crime insights. It is on this line that this Chapter focuses to explore ways in which frequent crime pattern mining can help in detecting useful patterns of crime from available datasets in the country. In this study, specific center of attention was on potential sources of crime data outside police stations. Swahili newspapers and online sources were employed for that purpose. Consequently, the Chapter’s specific objectives are threefold. To develop a pattern mining model and apply it to extract frequently reported crimes; to investigate on the distribution of the mined crimes per regions of the country; and to generate association rules between the mined crimes. Results from this study will be helpful to police and other law enforcement agencies in the process of crime detection and prevention in the country.

The remainder of this Chapter is organized as follows: section 2.2 reviews various related literature; section 2.3 presents the methodology used; section 2.4 presents results and discussions; section 2.5 concludes the Chapter.

2.2 Literature Review

In the first Chapter the concept of frequent patterns mining was briefly introduced. It was pointed out that, according to Han and Kamber (2006), a frequent pattern is a pattern or a set of items, subsequences or substructures that occurs in a dataset with frequency no less than a user-specified threshold. In other words, a frequent pattern can be defined as follows: given a set of items \( X = \{x_1, x_2, \ldots, x_n\} \), a set of transactions \( T = \{t_1, t_2, \ldots, t_m\} \), and \( S \) a subset of \( X \). \( S \) is a frequent pattern if it occurs in a percentage of all transactions in \( T \) that are not less than the minimum user-specified threshold.

Mining frequent patterns leads to the discovery of interesting associations and many other relationships among data. Association rules mining is defined by Han et al. (2000) as the method for discovering interesting relations between sets of items or objects in transaction databases, relational databases, and other information repositories. There are useful relationships that ex-
ist amongst the mined frequent itemsets. Association rule generation helps to uncover those relationships.

Apart from discovering interesting associations, mined frequent patterns can also be helpful in data classification, clustering, and other data mining tasks. Thus, frequent pattern mining has grown into a significant data mining task and a focused subject in several data mining studies (Han et al., 2000).

Although the idea of mining association rules originated from the analysis of market-basket data (it is further explained in Sections 3.1 and 3.2.1 of Chapter Three), association rules are not restricted to dependency analysis in the context of retail applications but are successfully applicable to a wide range of business problems to support effective decision making. Researchers have applied association rules into diverse areas, such as medical diagnosis to identify the probability of illness in a certain disease (Serban et al., 2006), enhance understanding about protein sequences (Mishra et al., 2006), analysis of census data to forecast public services planning (Brito and Malerba, 2003) and credit card fraud detection (Sánchez et al., 2009).

Frequent pattern and association rules mining have also become a powerful technique with great potential to help criminal investigators focus on the most important information on crime datasets. As such they help police investigating officers to identify hidden patterns from crime data (Varghese et al., 2010). A great deal of scientific research have, accordingly, been performed on crime patterns mining (Gangavane et al., 2015). Several of such research, however, have been concentrated on identifying crime patterns from crime databases and other structured data. For example, Zubi and Mahmmud (2014) proposed model for crime and criminal data analysis using data mining techniques. The authors used Libyan national criminal record data for their experiment which was based on association rule mining and clustering. Another research by Isafiade and Bagula (2013) which focused on creating a flexible and effective solution to crime situation recognition used crime incident reports that were stored on various crime databases. Wang et al. (2013a) proposed a pattern detection algorithm called Series Finder that grows a pattern of discovered crimes from within a database, starting from a seed of a few crimes. Series Finder used data collected by the Crime Analysis Unit of the Cambridge Police Department to detect patterns of crime committed by the same individual(s).
Same trend of generating crime patterns from structured crime data is observed in Jani (2014) and Varghese et al. (2010). Elyezjy and Elhaless (2015) attempted to investigate crime patterns using text mining and network analysis by mining offenders names from unstructured text data in the Arabic language. However, the authors did the mining from investigations documents that were obtained from police department. These, and other similar literature, show that there is little research in methods and techniques that extract frequent crime patterns from unstructured data in other sources outside police departments. There is, similarly, little research that consider the same extractions from datasets in local languages.

2.3 Research Methodology

2.3.1 Source of Data

Datasets that were used in this research were obtained from the following four reputable Swahili newspapers; Majira, Mtanzania, Mwananchi and Nipashe. The selected newspapers are published on daily basis. At least eight articles from each of the newspaper were collected and analyzed. The articles were those with crime related news reports published between April and May, 2016. Swahili newspapers were used because of two reasons. First, most of the newspapers in Tanzania are published in Swahili and second, the selected newspapers have news reporters from all over the country and hence countrywide coverage.

2.3.2 Workflow of the Mining Process

The first step in the process of crime patterns mining was articles collection. As it has already been pointed out, these articles were collected from the four newspapers. Collected articles were then preprocessed before fed to the patterns mining model that was built by using Rapid-Miner Studio. Preprocessing involved transforming the articles that were obtained from various newspapers’ platforms into a suitable format that the mining model could process.

The next step was ‘text processing’. This is actually what the patterns mining model was trained
to do. To accomplish this, the model reads and process document. In the ‘read document’,
the model loads preprocessed newspaper articles in .txt format. Several articles (of the same
newspaper) were then combined together as one document and taken to the ‘process document’
step.

The model was trained to do four things in the ‘process document’ step. First was ‘tokenize’ in
which words in the articles were grouped together and counted. Second was ‘filter stopwords’.
Stopwords are the most common words in a language. For example, stopwords in the English
language are such as the, is, at and which. RapidMiner has builtin dictionaries in several lan-
guages to find and filter stopwords out. Unfortunately, the articles that were used in this work
were in Swahili, the language which dictionary is not available in RapidMiner. Third step was
‘transform cases’. Since RapidMiner is case sensitive where letters that are uppercase do not
match with the same letters in lowercase, the study opted to use lower cases. So, in this step, all
letters were transformed into lower case.

The fourth and last step was ‘replace tokens’. Tokens are words, phrases, symbols or other
meaningful elements in the articles. Several of such tokens can be presented differently but
meaning the same thing. For example, the following were some of the tokens reported in Majira
newspaper; wauaji, kuuawa, kuwauwa, waliouawa, wameuawa, kifo, kuua, kumuua (which
literary means killers, being killed, killing them, those who were killed, are killed, death, to
kill, killing him), which all of them meant killing (mauaji) thus replaced by mauaji (killing) as
Fig. 4 shows. In this step, similar tokens were replaced by more common ones, and their total
occurences were recorded. In this way, the study was able to obtain various reported crimes,
their frequency of occurences, and the regions in which they occured. This mining process
workflow is summarized in Fig. 5.
Figure 4: Different tokens replaced by *mauaji*

Figure 5: Workflow of the proposed solution
2.3.3 Association Rules Generation

(i) Preparing Data for Rules Generation

After obtaining a set of mined crimes, FP-Growth algorithm was employed to generate association rules between those crimes. Since the mining process was done from four different newspapers, frequency of occurrences of the mined crimes varied from one newspaper to another. To make FP-Growth applicable in this case, the mined dataset was organized in tabular form in such a way that mined crimes were set as attributes of the table and the newspapers where the crimes were mined were set as rows, as shown in Fig. 6. A Boolean value ‘TRUE’ or ‘FALSE’ was assigned in each attribute to indicate whether a particular crime was reported in a particular newspaper or not.

![Sample data prepared for rules generation](image)

Figure 6: Sample data prepared for rules generation

(ii) Rules Generation Process

The prepared dataset was then loaded to RapidMiner for association rules generation. To accomplish rules generation ‘FP–Growth’ and ‘Generate Association Rules’ operators were used.
But due to the nature of the dataset the FP-Growth could not be applied directly because it requires all attributes to be binominal. Some preprocessing to mold the dataset into the desired form were done. In fact, after retrieving the prepared crimes dataset, ‘Text to Nominal’ and ‘Nominal to Binominal’ operators were used to preprocess the data before FP-Growth was used. This process is shown in Fig. 7.

Figure 7: Operators involved in the association rules generation process

The Text to Nominal operator was applied to convert all text attributes in the dataset to nominal attributes. After that conversion, Nominal to Binominal operator was applied to change those nominal attributes to binominal. FP-Growth operator was then applied to generate frequent itemsets. And, finally, to generate a set of association rules from the generated frequent itemsets the Create Association Rules operator was used. The aim was to generate only strong association rules between the mined crimes, so minimum support and confidence were set to be 0.95 and 1 respectively.

2.4 Results

2.4.1 Crime Occurrence Frequencies

Ten (10) crime incident types were extracted from the input files. Since the input files were in Swahili, the mined results were also in Swahili. The following are the results with their English translation in parentheses; Mauaji (Killings), Unyama (Brutality), Milipuko (Explosives), Makosa ya kingono (Sexual offenses), Ujambazi/Uvamizi (Invading/Gangs), Uhalifu wa kutumia Bunduki (Gunned crimes), Uhalifu wa kutumia Silaha za jadi (Traditional armed crimes), Ugaidi (Terrorism), Madawa ya kulevya (Drugs), and Mauaji ya Albino (Killing of people with albinism). Table 3, 4, 5 and 6 summarize this finding.
As shown in the presented results, *Mauaji* (Killings) occurred mostly in all of the four newspapers. Its frequency of occurrence was 60 in the 4 articles of Majira newspaper, 59 in 7 articles of Mtanzania, 98 in 8 articles of Mwananchi, and 36 in 6 articles of Nipashe. It can further be observed that other crime incident types with high frequency of occurrence were; *Unyama* (Brutality), *Milipuko* (Explosives) and *Ujambazi/Uvamizi* (Invading/Gangs).

In the other hand, *Mauaji ya Albino* (Killing of people with albinism) were seldom reported. It appeared only in Mwananchi newspaper, which tells that in May, 2016 there were very few incidents related to killings of people with albinism in Tanzania. The next least reported criminal incident was *Madawa ya kulevya* (Drugs), which was not reported at all in Majira newspaper.
Table 3: Mined crimes from MAJIRA newspaper (April/May, 2016)

<table>
<thead>
<tr>
<th>Mined Crimes</th>
<th>Total Frequency</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauaji</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>Unyama</td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>Milipuko</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Makosa ya kingono</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Ujambazi/Uvamizi</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>Uhalifu wa kutumia Bunduki</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Uhalifu wa Silaha za jadi</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Ugaidi</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Madawa ya kulevya</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mauaji ya Albino</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4: Mined crimes from MTANZANIA newspaper (April/May, 2016)

<table>
<thead>
<tr>
<th>Mined Crimes</th>
<th>Total Frequency</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauaji</td>
<td>59</td>
<td>7</td>
</tr>
<tr>
<td>Unyama</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Milipuko</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Makosa ya kingono</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Ujambazi/Uvamizi</td>
<td>35</td>
<td>8</td>
</tr>
<tr>
<td>Uhalifu wa kutumia Bunduki</td>
<td>27</td>
<td>4</td>
</tr>
<tr>
<td>Uhalifu wa Silaha za jadi</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>Ugaidi</td>
<td>45</td>
<td>2</td>
</tr>
<tr>
<td>Madawa ya kulevya</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Mauaji ya Albino</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 5: Mined crimes from MWANANCHI newspaper (April/May, 2016)

<table>
<thead>
<tr>
<th>Mined Crimes</th>
<th>Total Frequency</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauaji</td>
<td>98</td>
<td>8</td>
</tr>
<tr>
<td>Unyama</td>
<td>17</td>
<td>6</td>
</tr>
<tr>
<td>Milipuko</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Makosa ya kingono</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Ujambazi/Uvamizi</td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>Uhalifu wa kutumia Bunduki</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>Uhalifu wa Silaha za jadi</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>Ugaidi</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Madawa ya kulevyva</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Mauaji ya Albino</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

### Table 6: Mined crimes from NIPASHE newspaper (April/May, 2016)

<table>
<thead>
<tr>
<th>Mined Crimes</th>
<th>Total Frequency</th>
<th>Number of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauaji</td>
<td>36</td>
<td>6</td>
</tr>
<tr>
<td>Unyama</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>Milipuko</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Makosa ya kingono</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>Ujambazi/Uvamizi</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>Uhalifu wa kutumia Bunduki</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Uhalifu wa Silaha za jadi</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>Ugaidi</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Madawa ya kulevyva</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mauaji ya Albino</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Another interesting observation that was revealed from experimental results was on how often the mined crimes were reported in different articles of the same newspaper. Figure 8 shows the number of newspaper articles that reported the mined crimes. As it can be seen in Fig. 8, *Killing, Envading/Gangs* and *Traditional armed crimes* were leading in terms of being reported in many different articles.

![Figure 8: Number of newspapers’ articles that reported the crimes](image)

### 2.4.2 Crime Occurrence by Regions

Investigating crimes distribution per regions was a second objective of this part of the study. Results showed that out of 30 regions of Tanzania, newspapers reported crime occurrences in 16 regions. Zanzibar islands consist of five regions, but for the purpose of this study the islands were treated as a single region, that is, the region of Zanzibar. Figure 9 shows how crimes were distributed across the regions of Tanzania.

As shown in Fig. 9, Mwanza region was leading by having an average criminal incidents of more than 10. Tanga followed with an average of 6 to 10 criminal incidents. Following these results, the two regions were categorized as relatively high crime zones in May, 2016. Consequently, the Tanzania Police Force and other law enforcement agencies and stakeholders including the general public could consider placing extra efforts to reduce the high crime rates in the regions.
14 regions; four in the lake zone (Kagera, Geita, Shinyanga and Mara), two northern regions (Arusha and Kilimanjaro), two coast regions (Dar es Salaam and Pwani), two central regions (Dodoma and Morogoro), and three southern highlands regions (Rukwa, Mbeya and Njombe) and Zanzibar had an average of one to five criminal incidents. This made these regions to be categorized as low crime zones.

The four newspapers that were used in this study did not report any crime incident in southern regions (Ruvuma, Lindi and Mtwara), western regions (Kigoma, Katavi and Tabora) as well as Simiyu, Singida, Manyara and Iringa. Basing on this finding it can be concluded that those regions were relatively safe in the time the news were reported.
2.4.3 Association Rules Generation

As Fig. 10 shows in a graphical depiction, six association rules were generated among the crimes that were mined from newspapers. These rules are further elaborated in the consequent textual descriptions.

![Graphical visualization of generated association rules]

**Figure 10:** Graphical visualization of generated association rules

[Unyama] --> [Mauaji] (confidence: 1.000)
[Uhalifu wa kutumia Bunduki, Unyama] --> [Mauaji] (confidence: 1.000)
[Ugaidi, Unyama] --> [Mauaji] (confidence: 1.000)
[Milipuko, Unyama] --> [Mauaji] (confidence: 1.000)
[Uhalifu kwa silaha za jadi, Unyama] --> [Mauaji] (confidence: 1.000)
[Uhalifu kwa kutumia Bunduki, Milipuko, Unyama] --> [Mauaji] (confidence: 1.000)

What can be inferred from the generated rules is that since all the generated association rules concluded to *Mauaji* (Killings) then the premises (i.e. *Unyama* (Brutality), *Uhalifu wa kutumia Bunduki*, etc.) must be present.
*bunduki* (Gunned crimes), *Milipuko* (Explosives) and *Uhalifu kwa silaha za jadi* (Traditional armed crimes) possibly resulted into killings. Therefore, it is fair to say that killings resulted from several other crimes, and hence if brutality, gunned crimes, explosives and traditional armed crimes are contained, killings will also be contained. The mined patterns can be very useful to police and other law enforcement agencies in the cause of implementing their crime detection, prevention and control tactics.

Considering the patterns that were mined it is not surprising to see what have been reported by other researchers about the rise of crime fear. In fact, this research confirms the existence of crime incidents which are the contributing factors to the reported fear.

### 2.5 Effects of Increasing Dataset Size on Execution Time

On the technical aspect, the study went further to investigate on how the developed crime patterns mining model behaves with the increasing size of datasets. Unfortunately, the size of the dataset that were obtained from newspapers since were collected in a small period of time due to time constrains were small to be able to establish sensible results. Therefore, for this purpose, bigger size datasets were downloaded from two open-data websites; http://data.denvergov.org/download/gis/crime/csv/crime.csv and https://catalog.data.gov/dataset?tags=crime.

Crude dataset that were collected consisted of numerous attributes (as shown in Fig. 11), but specific interest was only on the ‘types of crime’ and ‘locations’ in which those crimes occurred (i.e. INCIDENT_ADDRESS). Therefore, data cleaning and pre-processing were done to obtain desired clean data (Fig. 12) to be used for experimentation purpose.

Cleaned and pre-processed dataset was then split into five separate files of varying sizes so as to evaluate the effects of the varying dataset sizes on the model’s execution time. Table 7 shows a list of dataset sizes with their corresponding number of records that we used.

Each of the five sets of cleaned data were fed into our patterns mining model in which we were able to generate extract frequent itemsets and generate association rules among the frequent
Table 7: Different dataset sizes used

<table>
<thead>
<tr>
<th>Datasets used</th>
<th>Size (in MB)</th>
<th>Number of records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>5</td>
<td>54,100</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>20</td>
<td>228,500</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>35</td>
<td>381,100</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>50</td>
<td>543,000</td>
</tr>
<tr>
<td>Dataset 5</td>
<td>100</td>
<td>1,048,575</td>
</tr>
</tbody>
</table>

itemsets in each of the five sets of data. Fig. 13 is an example of association rules generated from a 100 MB dataset.

In connection, time for completing execution of each of the dataset was recorded. Five MB dataset completed execution at 4 seconds, 20 MB completed at 11 seconds, 35 MB completed at 20 seconds, 50 MB completed at 27 seconds and 100 MB completed at 56 seconds. Similarly, we observed an increasing time for running FP-Growth operator with the increased dataset sizes. The algorithm executed the 5 MB dataset for 2 seconds, 20MB for 6 seconds, 35 MB for 8 seconds, 50 MB for 16 seconds, and 100MB for 35 seconds. Figure 14 shows these execution times.

These experiment results tell that, as size of the dataset increase do so execution time. So, if the
dataset size is big enough (in GB, TB or higher) the mining model, and similarly the FP-Growth algorithm, may either take too long to finish the process or may fail to process.

2.6 Chapter Conclusion

The main contribution of this work is the extraction of crime patterns from unstructured data in Swahili newspapers. By employing the frequent patterns mining model several frequently occurring crimes were extracted and association rules between the extracted crimes were generated. Distribution of crime occurrences per regions of Tanzania were also indicated. Mined
patterns can help police officers and other law enforcement agencies to understand the crime situation and thus put in place more efficient proactive measures against future crimes.
CHAPTER THREE

MINING FREQUENT PATTERNS OF CRIME USING MULTIPLE MINIMUM SUPPORTS FP-GROWTH BASED ON SHANNON ENTROPY

FP-Growth is one of the most effective and widely used association rules mining algorithm for discovering interesting relations between items in large datasets. Unfortunately, classical FP-Growth mines frequent patterns by using single user-defined minimum support threshold. This is not adequate for real life applications such as crime patterns mining. If, for instance, minimum support is set too low, huge amount of crime patterns (including uninteresting patterns) may be generated, and if it is set too high lots of interesting patterns (including seasonal patterns) may be lost. This Chapter proposes the use of Multiple Item Support (MIS) thresholds instead of single user-defined minimum support to tackle the challenge. The study employed Shannon entropy method to develop an algorithm that automatically obtains MIS values from crime datasets and use them to extract frequent patterns. The proposed approach was tested on different sizes datasets and found to be effective in terms of ability to extract patterns, reduced execution time and memory use.

3.1 Introduction

Mining association rules is one of the key data mining tasks. It discovers interesting relations or association rules amongst items in large databases and other data repositories. The idea of discovering association rules begun from the analysis of market-basket data. Market-basket analysis is the process of investigating customer buying behaviours by discovering associations amongst the different items that customers place in their ‘shopping baskets’. Today, association rule mining has become a powerful technique with huge potential and wide applications in several domains. One of such domains is crime patterns analysis, in which crime analysts extract interesting association rules from crime datasets. Such rules help to discover trends.

in criminal behaviour and predict or anticipate future criminal activities (Isafiade et al., 2015; Chen et al., 2004; Matto and Mwangoka, 2017).

There are several algorithms for mining association rules. According to Kumbhare and Chobe (2014) and as it have been described in Chapter One; Apriori, Eclat and FP-Growth are the most widely used. Comparative studies among such algorithms (see for example Kumbhare and Chobe (2014), Isafiade and Bagula (2013); and Kaur and Aggarwal (2013)) indicate FP-Growth as the most efficient in terms of number of database scans, execution time and memory consumption. FP-Growth has also been used extensively in mining crime patterns (Isafiade and Bagula, 2013; Laporais and Brandao, 2014; Matto and Mwangoka, 2017, 2018).

Classical FP-Growth mines frequent patterns by using a single user-specified minimum support (abbreviated as \( \text{minsup} \) and symbolized as \( \xi \)) threshold (Hu et al., 2016). Research have, however, shown that using single minimum support for crime patterns mining is not adequate as it does not reflect the nature of crime items in the dataset. Crime datasets consist of items that, some of them, vary greatly in terms of frequencies of occurrence. Some items appear so frequently in the dataset while others appear rarely.

Thus, if a minimum support is set very low, huge amount of crime patterns (including uninteresting patterns) will be generated. According to Han et al. (2011) this is because if an itemset is frequent, each of its subsets is frequent as well. For example, a frequent itemset of length 50, such as \( \{x_1, x_2, \ldots, x_{50}\} \), contains \( \binom{50}{1} = 50 \) frequent 1-itemsets (that is, \( \{x_1\}, \{x_2\}, \ldots, \{x_{50}\} \)); \( \binom{50}{2} = 1,225 \) frequent 2-itemsets (that is, \( \{x_1, x_2\}, \{x_1, x_3\}, \ldots, \{x_{49}, x_{50}\} \)); and so on. The total number of frequent itemsets that of length 50 is thus approximately \( 1.126 \times 10^{15} \) as shown in equation (3.1). This is huge number of itemsets for a computer to work on. Note that, the study employed the formula \( \binom{n}{r} = \frac{n!}{(n-r)!r!} \) to obtain combinations of different itemsets.

\[
\binom{50}{1} + \binom{50}{2} + \cdots + \binom{50}{50} = 2^{50} - 1 \approx 1.126 \times 10^{15} \quad (3.1)
\]
On the other side, if minimum support is set too high many interesting patterns may be lost since some of crimes, e.g. killing of people with albinism in the case of Tanzania (research reveals a rising number of albino killing during general elections (see Makoye (2015) for example)), occur seasonally and thus rarely found in the dataset.

In connection to the varying frequencies of crime items in the dataset, crime data have continuously being generated, collected and stored. And as the amounts of data keeps growing two things happen. First, it becomes harder for the FP-Growth to effectively extract crime patterns from them, and second, it takes too long to complete the patterns extraction process. Chapter Two have shown how the increasing amounts or sizes of data affects execution of the FP-Growth algorithm. To tackle the challenge, in this Chapter a method to enhance the FP-Growth algorithm is proposed. The proposed method replaces the use of a single user-defined minimum support threshold with an aggregate function that automatically computes items multiple minimum supports basing on empirical analysis of the dataset. This approach is based on the Shannon Entropy method. In connection, an algorithm for identifying such multiple minimum support values basing on the proposed approach is developed.

The rest of this Chapter is organized as follows; Section 3.2 presents a survey of related literature. This section discusses various association rule-mining concepts, revisits the FP-Growth algorithm and reviews previous studies that attempted to improve the FP-Growth. Section 3.3 describes the proposed approach. Section 3.4 presents experimental results that are based on the proposed approach, and the Chapter is concluded in Section 3.5.

3.2 Related Literature

3.2.1 Association Rules Mining

Association rule mining problem is stated by Fournier-Viger et al. (2012) and Kiran (2011) as follows. Suppose \( I = \{i_1, i_2, i_3, \ldots, i_n\} \) and \( T = \{t_1, t_2, t_3, \ldots, t_n\} \) are sets of finite items and transactions respectively. Suppose \( DB \) is a transaction database that comprises set of items in \( I \). \( DB = \{T_1, T_2, T_3, \ldots, T_n\} \) where \( Ti(i \in [1 \ldots n]) \). Suppose \( X \) and \( Y \) are itemsets, an
itemset is a set of items in $I$ i.e. $X \subseteq I$ and $Y \subseteq I$. An association rule of itemsets $X$ and $Y$ is denoted as $X \rightarrow Y$. This is the relationship between $X$ and $Y$, given that $X$ and $Y$ are disjoint itemsets.

Support of an itemset $X$, represented as $Sup(X)$, is the fraction of transactions $T$, comprising $X$. Similarly, support of itemset $Y$, represented as $Sup(Y)$, is the fraction of transactions $T$, comprising $Y$.

\[
Sup(X) = \frac{\text{Number of Transactions comprising } X}{\text{Total Number of Transactions}} \quad (3.2)
\]

Support of an association rule $X \rightarrow Y$, denoted as $Sup(X \rightarrow Y)$, is the fraction of transactions comprising of both $X$ and $Y$.

\[
Sup(X \rightarrow Y) = \frac{\text{Number of Transactions comprising } X,Y}{\text{Total Number of Transactions}} \quad (3.3)
\]

Confidence of an association rule $X \rightarrow Y$, denoted as $Conf(X \rightarrow Y)$, is an indication of how often the rule have been found to be true. It is given as

\[
Conf(X \rightarrow Y) = \frac{Sup(X \cup Y)}{Sup(X)} \quad (3.4)
\]

Zeng et al. (2015) pointed out that for a given user-specified minimum support, $\text{minsup}$, if the itemset meets the condition $Sup(X) \geq \text{minsup}$, then itemset $X$ is regarded as frequent itemset and conversely itemset $X$ is regarded as infrequent itemset.

### 3.2.2 FP-Growth Algorithm

The FP-Growth is one of the most efficient association rule mining algorithms. The algorithm mines frequent itemsets without generating the candidates. Han et al. (2000) proposed FP-
Growth algorithm to overcome the multiple database scans and candidate generations of the Apriori algorithm (Agrawal and Srikant, 1994). According to Han et al. (2004) FP-Growth uses a divide-and-conquer strategy to mine frequent itemsets. It uses two steps. First, build an FP-Tree by condensing a transaction database into a compressed structure. And second, extract itemsets directly from the FP-Tree that was built in the first step.

**Step 1: Building an FP-Tree**

According to Han et al. (2004), the algorithm for FP-Tree construction requires two sorts of inputs; the transaction database (DB) or the dataset and the minimum support threshold. The general steps for the FP-Tree construction are as shown in the following algorithm.
Algorithm: FP-Tree Construction

Input: DB, minsup (ξ)

Output: FP-tree

Process:

1. Scan DB once. Discard infrequent items and collect the set of frequent items, F, (and their supports). Sort F in support-descending order of frequent items list (FList).

2. Scan DB again to construct FP-Tree. Create the root of an FP-Tree, T, and label it as “null”. For each transaction (Tr) in DB do the following:
   
   - Select the frequent items in Tr and sort them according to the order of FList. Let the sorted FList in Tr be [p | P], where p is the first element and P is the remaining list. Call insert tree([p | P], T).
   
   - The function insert tree([p | P], T) is performed as follows:
     
     If T has a child N such that N.item-name = p.item-name, then increment N’s count by 1
     
     Else
     
     Create a new node N, with its count initialized to 1, its parent-link linked to T, and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert(P, N) recursively.
The way FP-Growth works can be described by considering a transaction database, DB, in Table 8. DB has eight different crime items, i.e. crime1, crime2, crime3, crime4, crime5, crime6, crime7 and crime8 represented as C1, C2, C3, C4, C5, C6, C7 and C8 respectively. \( \text{minsup} \) threshold for this case is assumed to be 2 (i.e. \( \xi = 2 \)).

**Table 8: Dataset in the Transaction Database (DB)**

<table>
<thead>
<tr>
<th>Transaction ID (TID)</th>
<th>Sets of Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T101</td>
<td>C1, C2</td>
</tr>
<tr>
<td>T102</td>
<td>C1, C5, C6</td>
</tr>
<tr>
<td>T103</td>
<td>C3, C4</td>
</tr>
<tr>
<td>T104</td>
<td>C1, C2, C8</td>
</tr>
<tr>
<td>T105</td>
<td>C3, C4</td>
</tr>
<tr>
<td>T106</td>
<td>C1, C3</td>
</tr>
<tr>
<td>T107</td>
<td>C1, C2</td>
</tr>
<tr>
<td>T108</td>
<td>C5, C6</td>
</tr>
<tr>
<td>T109</td>
<td>C3, C4, C7</td>
</tr>
<tr>
<td>T110</td>
<td>C1, C2</td>
</tr>
<tr>
<td>T011</td>
<td>C1, C2</td>
</tr>
<tr>
<td>T012</td>
<td>C1, C3</td>
</tr>
<tr>
<td>T013</td>
<td>C1, C2</td>
</tr>
<tr>
<td>T014</td>
<td>C2, C5, C6, C7</td>
</tr>
<tr>
<td>T015</td>
<td>C3, C4</td>
</tr>
<tr>
<td>T016</td>
<td>C1, C2, C4</td>
</tr>
<tr>
<td>T017</td>
<td>C3, C4</td>
</tr>
<tr>
<td>T018</td>
<td>C1, C3</td>
</tr>
<tr>
<td>T019</td>
<td>C1, C2, C5</td>
</tr>
<tr>
<td>T020</td>
<td>C3, C4</td>
</tr>
</tbody>
</table>

According to the above presented algorithm first, a scan of DB collects F, the set of frequent items and the support of each of those frequent items. \( F = \{ C1:12, C2:9, C3:9, C4:7, C5:4, C6:3, C7:2, C8:1 \} \). Then sort F in support-descending order as FList., while discarding those items
whose support is less than minsup threshold. FList is the list of frequent items. Since C6, C7 and C8 has support less than 3 they will be discarded and thus FList = C1:12, C2:9, C3:9, C4:7, C5:4, C6:3, C7:2. Second, this order and the methods indicated in the FP-Tree construction algorithm to build an FP-Tree are used. Figure 15 is the constructed FP-Tree. Link is added to speed lookup and easy matching of the pointer to FP-Tree.

**Figure 15**: A complete FP-Tree for a crime database (DB)

**Step 2: Frequent Itemset Generation**

After completing construction of the FP-Tree, the following step of the FP-Growth algorithm is to extract frequent itemsets from the FP-Tree. According to Tohidi and Ibrahim (2010) the FP-tree is extracted by dividing the tree (or the compressed database) into sub-databases (conditional pattern base). And from those conditional pattern bases we find out pattern fragments (conditional FP-Tree) associated with each of the databases, and lastly do mining recursively on the tree. Table 9 shows the conditional pattern base, conditional FP-Tree and the mined frequent itemsets from the constructed FP-Tree in Fig. 15.

### 3.2.3 Studies to improve FP-Growth Algorithm

FP-Growth algorithm has been blamed to produce large number of conditional pattern base and consequently conditional FP-tree recursively in mining the frequent patterns (Xia et al., 2013).
Table 9: Mined Patterns (with $\xi = 2$)

<table>
<thead>
<tr>
<th>Items</th>
<th>Conditional Pattern Base</th>
<th>Conditional FP-Tree</th>
<th>Frequent Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7</td>
<td>{C3, C4:1}, {C3, C5, C6:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C6</td>
<td>{C1, C5:1}, {C5:1}, {C2, C5:1}</td>
<td>{<a href="">C5:3</a>}</td>
<td>{C5, C6:3}</td>
</tr>
<tr>
<td>C5</td>
<td>{C1, C2:1}, {C1:1}, {C2:1}</td>
<td>{<a href="">C1:2</a>, <a href="">C2:2</a>}</td>
<td>{C1, C5:2, C2,C5:2, C1,C2,C5:2}</td>
</tr>
<tr>
<td>C4</td>
<td>{C1, C2:1}, {C3:6}</td>
<td>{<a href="">C3:6</a>}</td>
<td>{C3,C4:6}</td>
</tr>
<tr>
<td>C2</td>
<td>{C1:8}</td>
<td>{<a href="">C1:8</a>}</td>
<td>{C1,C2:8}</td>
</tr>
<tr>
<td>C3</td>
<td>{C1:3}</td>
<td>{<a href="">C1:3</a>}</td>
<td>{C1, C2:7}</td>
</tr>
</tbody>
</table>

The illustration results in Table 9, which is based on a 20 transactions dataset, can be a good example on this. Studies have thus shown that the algorithm becomes less effective when the dataset size increases. Consequently, several methods have been suggested to improve efficiency of the algorithm. Some of such methods are; implementation of parallel FP-Growth (Li et al., 2008; Xia et al., 2013; Pramudiono and Kitsuregawa, 2003), mining only top-k frequent itemsets (Lee and Clifton, 2014; Wang et al., 2013b), and the use of distributed computing for frequent patterns mining (Deng and Lou, 2015; Itkar and Kulkarni, 2013). Although all of these, and other similar approaches, try to tackle the challenge of the algorithm especially with the increasing datasets, they ignore the mining for infrequent items. In fact, these techniques employ a single user specified minsup. The user specified minimum support threshold assumes that items in the dataset are of identical nature and occurrences. This is however a rare situation in real life applications especially in the crimes datasets where some crime items appear so regularly in the dataset while others appear rarely. It is on this same line that Isafiade et al. (2015) used a quartile floor-ceiling functions of the descriptive statistics to propose a pruning step of the FP-Growth. This approach automatically identifies the minsup threshold for the fine-tuning of the algorithms pruning step for identifying frequent crime pattern trends. Unfortunately this method works only for small datasets.
3.2.4 Approaches that use Multiple Minimum Support

To improve extraction of frequent itemsets studies have proposed the use of multiple minimum supports approach (see for example Bhatt and Patel (2015) and Chunjiang et al. (2007)). Liu et al. (1999) used this approach to mine rare itemsets through an Apriori-like algorithm called Multiple Support Apriori (MSApriori). According to the author, the approach assigns each item with a minimum support value known as ”Minimum Item Support” (MIS). Frequent itemsets are produced under the condition that they satisfy the lowest MIS value amongst the corresponding items.

In this multiple minimum support approach, association rules definition remains the same as presented in section 3.2.1 above, but the rule’s \( \text{minsup} \) is defined in terms of MIS of items occurred in the rule. In other words, each item in the database can have MIS value that is calculated using a formula or stated by the user. The provision of different MIS values for different items helps the user to efficiently define distinctive support needs for distinctive rules.

For instance, if a dataset consists of four crime items, e.g. murder, robbery, killing of albino, and rape, then MIS values could vary as follows: \( \text{MIS(murder)} = 3, \text{MIS(robbery)} = 5, \text{MIS(killing of albino)} = 0.1, \text{MIS(rape)} = 0.5. \) In addition, the minimum support for any itemset \( X = \{i_1, i_2, \ldots, i_k\}, 1 \leq k \leq n, \) is given as

\[
\text{minsup}(X) = \text{minimum}(\text{MIS}(i_1), \text{MIS}(i_2), \ldots, \text{MIS}(i_k))
\]

(3.5)

According to Liu et al. (1999) MIS for every 1-itemset (in the MSApriori), expressed as \( \text{MIS}(i) \), is calculated using the following percentage-based formula:

\[
\text{MIS}(i) = \begin{cases} 
M(i) & \text{if } M(i) > \text{LS} \\
\text{LS} & \text{otherwise}
\end{cases}
\]

(3.6)

\[M(i) = \beta \cdot f(i)\]

Where:
LS: least support, stated by the user to express the lowest allowed minimum item support, 
\( f(i) \): frequency of occurrence of an item in the dataset, and 
\( \beta \): value that governs how MIS values should be associated to their occurrences.

Unfortunately, MSApriori undergoes the same performance drawbacks as the classical Apriori algorithm (Han et al., 2004). FP-Growth-like algorithms that use multiple minsup were then proposed. Specifically, Hu and Chen (2006) proposed CFPGrowth algorithm, and later Kiran and Re (2009) proposed the CFPGrowth++ algorithm. According to Kiran (2011), the main idea of CFPGrowth++ was the use of the notion of Support Difference (SD) instead of a percentage-based methodology, to specify items’ MIS values as follows.

\[
MIS(i) = \max(\text{Sup}(i) - SD, LS) \quad (3.7)
\]

SD can be either user-specified or calculated from the formula \( SD = \lambda(1 - \beta) \), where, \( \lambda \) is a parameters such as mean, median, and mode of the item, and \( \beta \) and \( LS \) is the same as for MSApriori.

Although CFPGrowth++ have shown improvements as compared to its predecessors, studies identified that its main weakness is on stating ”good” MIS value for each item. According to Chen et al. (2014), for example, the algorithm requires users to identify a minimum support value for each item and continuously tune it to obtain the best value. This is a costly in terms of time and efforts.

3.3 The Proposed Approach

This study’s approach for multiple minimum supports FP-Growth is based on Shannon entropy (also known as Information Entropy). Entropy is simply the average (expected) amount of the information from the event. According to Lesne (2014) the Shannon entropy of X is given as;

\[
E(X) = - \sum_{i=1}^{n} P_i * \log_2 P_i \quad (3.8)
\]
Where:

\[ E(X) \] (sometimes denoted as \( H(X) \)): entropy of a random variable/item \( X \),

\( n \): number of different outcomes, and

\( P_i \): probability of a given item

The Shannon entropy equation (3.8) is used to obtain entropy of each of the crime items in the crime dataset basing on the frequency of occurrence of each of those items. To reflect this study’s context, the Shannon entropy equation is rewritten as shown in equation (3.9) below. In this equation, \( C \) represents crime item.

\[
E(C) = -\sum_{i=1}^{n} P(C_i) \log_2 P(C_i)
\]  

(3.9)

This gives the probability of occurrence of a particular crime from a set of similar crime items. In this case, when the number of crime items increase the probability decreases, and thus the entropy. In other words, highly occurring crimes will have higher entropy than the low occurring crimes. To avoid this situation reciprocal of the entropy is taken. Reciprocal of the entropy assigns entropy values that increase with the increase of frequency of occurrence of an item. Our entropy equation for crime items thus becomes as shown in equation (3.10).

\[
E(C) = \left[ -\sum_{i=1}^{n} (P(C_i) \log_2 P(C_i)) \right]^{-1}
\]  

(3.10)

The entropy value obtained in equation 3.4 above gives the \( MIS \) values of crime items in the dataset. Thus, equation 3.4 is rewritten in terms of \( MIS \). In this case \( E(C) \) is replaced with \( MIS(C_i) \) to obtain equation (3.11).

\[
MIS(C_i) = \left[ -\sum_{i=1}^{n} (P(C_i) \log_2 P(C_i)) \right]^{-1}
\]  

(3.11)

Where: \( MIS(C_i) \) is the minimum item support of crime item \( i \) when \( MIS(C_i) \) is greater than or equals to \( LS \), otherwise \( MIS(C_i) = LS \). As Liu et al. (1999) defines, \( LS \) is the user-specified Least Support. The final \( MIS \) of an item will not entirely depend on the value obtained from
our aggregate function in equation (3.11). Depending on the nature of dataset, calculated MIS value can even be one or less than one. We use the concept of $LS$, where user will set the least support value, to avoid the possibility of getting unreasonable $MIS$. Pseudocode on how to obtain $MIS$ by using this approach is shown in Algorithm (in Table 10). Source code for the implementation of this algorithm is appended in Appendix 6.

**Table 10:** The proposed algorithm for specifying MIS values

<table>
<thead>
<tr>
<th>Algorithm: Specifying MIS values based on Shannon Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Transaction database (DB), Least Support ($LS$).</td>
</tr>
<tr>
<td><strong>Output:</strong> Complete set of MIS values</td>
</tr>
<tr>
<td><strong>Process:</strong></td>
</tr>
<tr>
<td>1. Scan DB; Count the total number of available distinct crimes in each of the crime categories found in DB. Call the counts $N(C_i)$.</td>
</tr>
<tr>
<td>2. For every crime item ($C_i$) compute the probability of $C_i$ ($P(C_i)$) as $\frac{1}{N(C_i)}$</td>
</tr>
<tr>
<td>3. Compute the entropy of crime item ($C_i$) as $E(C_i) = -(P(C_i) \ast \ln(P(C_i)))$</td>
</tr>
<tr>
<td>4. Compute reciprocal ($(E(C_i))^{-1}$) of the entropy obtained in 3 above</td>
</tr>
<tr>
<td>5. <strong>If</strong> $(E(C_i))^{-1} \geq LS$ <strong>then</strong></td>
</tr>
<tr>
<td><strong>then</strong> $MIS(C_i) = (E(C_i))^{-1}$</td>
</tr>
<tr>
<td><strong>else</strong> $MIS(C_i) = LS$</td>
</tr>
</tbody>
</table>

The calculated MIS values are then used to obtain the conditional pattern base and conditional FP-tree from the FP-tree. FP-tree construction uses the same procedures as shown in Algorithm 1 above, but in this case $\xi$ is $LS$. For example, suppose $LS = 2$, basing on the transaction database in Table 3.1 and the constructed FP-tree in Fig. 3.1, the obtained MIS values, conditional pattern base, conditional FP-Tree and the mined frequent itemsets will be as shown in Table 11.
Table 11: Mined Patterns (with the proposed MIS approach, $LS = 2$)

<table>
<thead>
<tr>
<th>Items</th>
<th>MIS</th>
<th>Conditional Pattern Base</th>
<th>Conditional FP-Tree</th>
<th>Frequent Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7</td>
<td>3</td>
<td>{C3, C4:1}, {C3, C5, C6:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C6</td>
<td>3</td>
<td>{C1, C5:1}, {C5:1}, {C2, C5:1}</td>
<td>{<a href="">C5:3</a>}</td>
<td>{C5, C6:3}</td>
</tr>
<tr>
<td>C5</td>
<td>3</td>
<td>{C1, C2:1}, {C1:1}, {C2:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C4</td>
<td>4</td>
<td>{C1, C2:1}, {C3:6}</td>
<td>{<a href="">C3:6</a>}</td>
<td>{C3,C4:6}</td>
</tr>
<tr>
<td>C2</td>
<td>4</td>
<td>{C1:8}</td>
<td>{<a href="">C1:8</a>}</td>
<td>{C1,C2:8}</td>
</tr>
<tr>
<td>C3</td>
<td>4</td>
<td>{C1:3}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

3.4 Experimental Results

In order to verify effectiveness of the proposed solution several experiments on crime datasets with varying sizes were conducted. The experiments were run on an Intel core i5-4570 3.40 GHz processor machine with 8GB of memory, and Microsoft Windows 7 operating system. Java language was used to write codes for the proposed solution.

Experimentation datasets used in this part of study was obtained online from the link https://catalog.data.gov/dataset?tags=crime. After obtaining the datasets cleaning and pre-processing were done to filter unnecessary and irrelevant data, and sometimes filling the missing values. In fact, the dataset consisted of both criminal and traffic offences, but our interest was only on the criminal offences. The obtained data were grouped into four groups. The first group was 5KB with 847 records (this dataset is represented as DATASET-I, the second was 10KB with 1390 records (represented as DATASET-II), third was 15KB with 2162 records (represented as DATASET-III), and the fourth was 20 KB with 2910 records (represented as DATASET-IV). We then tested our solution on each of these datasets to see how it behaves on the varying dataset sizes. Evaluation criteria were execution time and memory use. Thus, there was a comparison of execution time and memory consumption on our proposed solution against classical FP-Growth with varying minimum support thresholds.

Table 12 shows memory consumption in the classical FP-Growth with minimum supports of 10, 20 and 30, and memory consumption with the proposed solution. It was observed that
varying user-defined minimum supports did not affect memory consumption of the FP-Growth algorithm, but when the size of the dataset increased the proposed solution in this study was more effective in terms of memory consumption.

Table 12: Memory Use in MB (Proposed Approach vs. FP-Growth)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MinSup=10</th>
<th>MinSup=20</th>
<th>MinSup=30</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATASET-I</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>DATASET-II</td>
<td>2.24</td>
<td>2.24</td>
<td>2.24</td>
<td>2.24</td>
</tr>
<tr>
<td>DATASET-III</td>
<td>2.88</td>
<td>2.88</td>
<td>2.88</td>
<td>2.56</td>
</tr>
<tr>
<td>DATASET-IV</td>
<td>3.52</td>
<td>3.52</td>
<td>3.52</td>
<td>2.86</td>
</tr>
</tbody>
</table>

Concerning execution time, as shown in Fig. 16 an increase of execution time was observed as the dataset increased. The proposed solution recorded a lower execution time as compared to the classical FP-Growth algorithm.

Figure 16: Execution Time in ms (Proposed Approach vs. FP-Growth)

Apart from observing the running time and memory use of the proposed solution against classical FP-Growth with varying minimum supports, the study went further to observe its running time and memory use against CFPGrowth algorithm. The study compared running time and memory use of the proposed solution against that of CFPGrowth because CFPGrowth is an FP-Growth based algorithm that extracts frequent patterns using multiple minimum supports.
approach same as the proposed solution. For the purpose of these experiments six sets of varying sizes datasets (i.e. datasets with 500, 1000, 1500, 2000, 2500 and 3000 rows) were used.

Experimental results on both run time and memory use show that the proposed approach yields better run-time and memory use in comparison with CFPGrowth algorithm. Figure 17 and 18 show run time of and memory use respectively of the two approaches for similar sets of crime data.

**Figure 17:** Execution Time in ms (Proposed Approach vs. CFPGrowth)

**Figure 18:** Memory Use in MB (Proposed Approach vs. CFPGrowth)
3.5 Chapter Conclusion

In this Chapter, an effective approach for handling the rare item problem in crime datasets through the use of an FP-Growth with Multiple Itemset Support (MIS) values was proposed. The proposed algorithm, which is based on the Shannon Entropy, automatically scans the entire dataset and assigns MIS values on each crime item in the dataset basing on how frequently it has occurred in the given dataset. In this way, the proposed approach handles not only those crime items that rarely appear in the dataset but also those that appear very frequently. In other words, the proposed approach handles each crime item separately depending on how frequently it has appeared in the dataset. The approach was tested on varying crime datasets and compare execution time and memory consumption of the solution with the varying minimum supports FP-Growth as well as the CFPGrowth algorithm. Experimental results show that the proposed solution is reasonable and more effective as it has shown better performance in terms of execution time and memory usage.
CHAPTER FOUR

FRAMEWORK FOR EXTRACTION OF FREQUENT PATTERNS OF CRIME FROM MULTIPLE SOURCES OF DATA USING ENTROPY BASED FP-GROWTH ALGORITHM

Mining of frequent crime patterns have remained under a substantial study over the past few years. Consequently, several scholars have proposed frameworks on which the pattern mining works can be achieved. Unfortunately, most of the existing patterns mining frameworks are faced with some similar challenges; lack of consideration of multiple sources of input data, lack of clear guideline on how input data is cleaned and pre-processed, and lack of consideration on how the rare item problem in crime datasets is to be dealt with. This Chapter presents a proposed framework that uses the multiple minimum supports FP-Growth algorithm based on the proposed FP-Growth enhancing method presented in section 3.3 of Chapter Three to overcome the identified challenges in the existing frameworks.

4.1 Introduction

As it have been pointed out in the previous Chapters, there are many algorithms and techniques for extracting frequent itemsets and generating association rules. Research has shown FP-Growth as the most popularly used. Unfortunately, the algorithm suffers rare item problem especially when the amount of data increases. In this regard, the use of a multiple minimum support FP-Growth that is based on the Shannon entropy as an approach to enhance the algorithm is proposed. In connection to having different techniques and algorithms for frequent patterns mining, research have also proposed several frameworks for crime patterns mining (Nasridinov et al., 2016; Tayal et al., 2015; Iqbal et al., 2012).

Most of the existing frameworks, however, either consider extraction of patterns from single source of crime data or do not put consideration for any input data source. But, as we have
previously explained, crime data comes from different sources (e.g. crimes databases, news articles, and social media, among others). These sources provide data with varying quantities and structures and thus may require different methods for pre-processing of data and extraction of patterns. In this case, a framework for mining crime patterns from structured data for example, cannot be efficiently employed for unstructured data. And so, if the mining task is to involve both structured and unstructured data then more than one framework have to be employed, which is inconvenient and time consuming.

This Chapter proposes a generic framework that provides input data flexibility. The framework uses a multiple minimum supports FP-Growth algorithm that is based on the Shannon Entropy to extract frequent crime patterns from multiple sources of crime data.

The remainder of this Chapter is organized as follows: Section 4.2 revisits some existing works that proposed similar frameworks and identifies the existing challenges that this work is put forward to address. Section 4.3 describes the proposed framework, Section 4.4 presents experimental results on the proposed framework, and Section 4.5 concludes the Chapter.

4.2 Related Works

Mining of patterns of crime from crime datasets have remained under a significant study over the past few years. The increased generation of data in recent years has boosted its importance. Consequently, several scholars have proposed frameworks on which the pattern mining works can be achieved. Keyvanpour et al. (2011), for example, introduced a multi-purpose framework for intelligent crime investigation. The framework used a systematic approach for using neural networks (i.e. Self-Organizing Map (SOM) and Multilayered Perceptron (MLP)) for clustering and classifying crime data. The framework is furthermore based on data mining techniques to discover key entities from plain-text written police narrative reports. Similarly, Chen et al. (2004) developed a crime data mining framework by combining data mining techniques such as link analysis, association rule mining, and knowledge discovery. The framework has the capability of identifying different kinds of crimes and their associations on the police department database.
Another work by Iqbal et al. (2012) introduced a framework for extracting criminal information and forensically relevant information from skeptical online messages. The framework takes online messages as input, extracts the social networks from the log, summarizes chat conversations into topics, identifies the information relevant to crime investigation, and visualizes the knowledge for an investigator. Nasridinov et al. (2016) proposed a data mining framework for predicting crimes. The proposed framework consists of the following modules: test data generation, classification, clustering and data ranking. Specifically, the authors used kNN, k-means and Skyline algorithms. This framework did not consider the type of input data. The work by Tayal et al. (2015) proposed this same kind of framework.

Most of the existing frameworks, as have just been outlined, are faced with some similar challenges. The most observable one is that many of such frameworks either consider only one source of input data (e.g. narrative crime reports from police (Keyvanpour et al., 2011), police department database (Chen et al., 2004), chat logs (Iqbal et al., 2012), etc.), or do not consider any input source of data (Nasridinov et al., 2016; Tayal et al., 2015). This is considered as a challenge because of two reasons. First, patterns of crime can be mined from more than one data source which creates a need for a framework that provides input data flexibility. Second, the type of input data normally determines the choice of the mining method, thus if the type of input data is not known it makes it difficult to decide on the type of mining method(s) to be employed. Another challenge associated with existing frameworks is that most of the frameworks were introduced with no clear guideline on how the input data are to be cleaned and pre-processed, and how the rare item problem in crime datasets is to be dealt with.

4.3 The Proposed Framework

In order to overcome the identified weaknesses in the existing frameworks, this research proposes a generic framework that extracts patterns of crime from multiple data sources. The framework consists of four main stages: data collection (or data sources), data cleaning and pre-processing, patterns generation, and patterns evaluation and visualization as shown in Fig. 19 and further described in subsequent sections.
4.3.1 Data Sources

The first stage of the proposed framework involves collection of data in which patterns are to be extracted. Since crime data can be obtained from different sources, four sources have specifically been considered as follows: crimes and criminals databases, crime reports, news articles and social media. We further describe each of these sources and how they can provide crime data.

(i) Crimes and criminals databases: Most of police departments have databases of crimes. Such databases consist of detailed information about reported crimes, such as type of crime, how, when and where it was committed, who committed it and why. In connection to having crime databases, police departments keep various records of criminals such as names, age, gender, marital status, fingerprints, and photographs, among others (Asege-
hgn, 2003). Once these data are analyzed they can give useful insights necessary for crime preventions.

(ii) **Crime reports**: Government organs such as Bureau of Statistics as well as non-government agencies publish several crime related reports for public use. Those publications or reports are considered as potential sources of crime data. In fact, a compilation of many of such reports once analyzed can provide useful crime trends.

(iii) **News articles**: News media such as newspapers, blogs, and informational websites report several sorts of crime that happen in communities. Unlike crimes and criminal databases that are mainly based on reported crimes, reports by news articles are based on both reported and unreported crimes. The articles can be analyzed to provide useful patterns and trends of crime. Several research works, such as Bais and Adhikari (2017) and Iglesias et al. (2016), have also attempted to extract such crime patterns from news articles.

(iv) **Social Media**: Social media has become one of the most popular platforms to allow users to communicate, and share their views and interests. With the rapid growth of social media sites such as Facebook, Instagram, Twitter, and WhatsApp there is vast amount of user-generated contents. Some of those contents once analyzed can help to provide useful crime insights. Iqbal et al. (2012) and Chan and Liebowitz (2005) have also attempted to mine crime patterns from social media.

### 4.3.2 Data Cleaning and Pre-processing

After data has been collected the next stage involved cleaning and pre-processing such data. This stage involves filtering unnecessary and irrelevant data, filling the missing values, and transforming the data into suitable format for the mining task. Steps involved in cleaning and pre-processing of data depends on the type of input data as shown in Fig. 19.

Structured data, for example, comes from databases in which missing values are a common occurrence. So, as part of cleaning and pre-processing of such data this framework do not ignore or remove the missing values but rather it replace them with other values. This process is known as imputation of the missing values. Concerning unstructured data, the framework do
not impute the missing values because the data is naturally unstructured. Instead, steps involved in unstructured data are as follows; tokenization (grouping together and counting words in the data file), transform cases (changing everything in the data file to either lower or upper case), stemming (in this step the words are reduced into their stems (also known as roots). Since the meaning of different words could be the same but their form different, it is necessary to identify each word using its stem form), and filtering stopwords (eliminating stopwords from the data file. Stopwords are the most common words such as prepositions, and pro-nouns that are unlikely to help in the mining task).

Since this framework is based on the FP-Growth algorithm, which works with discrete data, and since the input data is not necessarily discrete, discretization is also involved as part of data pre-processing. According to Karthikeyan and Vembandasamy (2015) discretization is an essential originator to using association rule mining algorithms. This work adopted ChiMerge discretization method to achieve the process.

The ChiMerge is a supervised global discretization method introduced by Kerber (1992). It is one kind of discretize methods intended for discretizing the data and it utilizes Chi Square statistics that is efficient on the attributes of a record. The neighboring pairs of values are evaluated to discover the match among the data using chi square analysis. If the data are comparable then they are reserved in same interval and if not then they are put in different intervals. The method uses $X^2$ test to determine whether the current point is merged or not. The formula for computing $X^2$ value is as shown in equation (4.12).

$$X^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

(4.12)

Where:

$k$: number of system classes

$A_{ij}$: number of patterns in $i^{th}$ interval, $j^{th}$ class

$C_j = \sum_{i=1}^{2} A_{ij}$ : number of patterns in $j^{th}$ class
\[ R_i = \sum_{j=1}^{k} A_{ij} \]: number of patterns in \( i \)th interval

\[ N = \sum_{i=1}^{2} R_i \]: total number of patterns

\[ E_{ij} = \frac{R_i X C_j}{N} \]: expected frequency (entropy) of \( A_{ij} \)

### 4.3.3 Pattern Generation

In this step, sets of items that occur frequently in the dataset are discovered. Frequent items are those items whose support is greater or equal to the specified minimum support threshold \((\text{minsup})\). In other words, item \( X \) is frequent if \( \text{support}(X) \geq \text{minsup} \). FP-Growth algorithm is used to generate frequent patterns from data. The reasons for using this algorithm have already been explained in section 1.1 of Chapter One. FP-Growth works by compressing the input data into a prefix tree known as FP-tree and then extract frequent itemsets from the constructed FP-tree.

However, because of the nature of crime data the classical FP-Growth may suffer the rare item problem and thus may not be adequate. To confront the challenge, this framework uses the multiple minimum support FP-Growth. In this case, the \( \text{minsup} \) for any crime itemset \( X \) is represented with the minimal Minimum Item Support (MIS) value among all items, as shown in equation (4.13).

\[
\text{minsup}(X) = \min(MIS(i_1), MIS(i_2), \ldots, MIS(i_k)) \tag{4.13}
\]

Where:

\[ X = i_1, i_2, \ldots, i_k, 1 \leq k \leq n, \] is a pattern, and

\[ MIS(i_j), 1 \leq j \leq k, \] represents the \( MIS \) of an item \( i_j \in X \)

Several FP-Growth based algorithms that use this concept of multiple minimum supports have been introduced. CFPGrowth and CFPGrowth++ proposed by Hu and Chen (2006) and Kiran (2011) respectively are some of them. These algorithms are, however, faced with a challenge
of how to obtain good MIS values. So, instead of using these algorithms as they are, the study have proposed an effective procedure for obtaining MIS values. The procedure is based on the Shannon Entropy method and it assigns MIS values to the items depending on how frequently they have appeared in the dataset. Flowchart on how this works is shown in Fig. 20 and the source code for its implementation is as appended in Appendix 6.

Figure 20: Flowchart for obtaining MIS values

The obtained MIS values are then used to extract frequent itemsets and generate association rules. The process of extracting frequent itemsets and generating association rules are the same
as those of the FP-Growth with an exception that in this framework, the user defined minimum support is replaced by the presented procedure.

4.3.4 Pattern Evaluation and Visualization

Pattern evaluation and visualization is the final stage of this framework. In patterns evaluation an investigation of the results generated in previous step is done. The aim is at eliminating irrelevant rules or patterns and remain with only relevant ones. As a result, this step may take the user back to the data cleaning and pre-processing step where he can further clean the data so as to obtain significant results. Patterns visualization involves presenting extracted patterns in the forms that make them easily understandable to users. This step may involve the use of textual descriptions or visual aids such as graphs and charts. In fact, this stage is an essential part for law enforcement agencies and other stakeholders to visualize the mined patterns so as to make necessary interventions.

4.4 Framework Validation

This framework was proposed following a series of experiments that were conducted in Chapter Two and Three. In those chapters experiments that involved both structured and unstructured data were conducted. Chapter Two, for example, presented the patterns mining model that extracts crime patterns from unstructured data, the news articles. The process in the extraction of such patterns are summarized in the framework. Chapter Three was concerned with structured data obtained from online source and crime/criminal database. Crime patterns extracted in these experiments were obtained following a series of steps as presented in the proposed framework.

Although the processes involved in such experiments partly validates the framework, Chapter Five performs further evaluation and validation. However, more experimentations can be conducted on more other sources of data to further validate the framework. Chapter Five presents a prototype that is basically involved with extraction of crime patterns from structured datasets. Implementation of this prototype validates the framework on the part of patterns extraction from
structured data.

4.5 Chapter Conclusion

This Chapter have proposed a framework for extraction of pattern of crime from multiple sources of crime data. The framework is proposed to confront the challenges such as lack of input data flexibility, lack of clear procedures for cleaning and pre-processing the input data, as well as the rare item challenge in crime datasets that were observed in the existing frameworks. The framework confronts these challenges in a process consisting of four stages. First stage is data collection (or data source) that allows flexibility for both structured and unstructured data to be involved. Second stage is data cleaning and pre-processing. Depending on the type of input data, cleaning and preprocessing may involve replacing missing values, tokenizing, transforming cases, stemming, filtering stopwords and discretization. Third stage is pattern generation. In this stage patterns are generated by using a multiple minimum support FP-Growth algorithm based on the Shannon Entropy. Last stage is pattern evaluation and visualization in which generated patterns are investigated to eliminate irrelevant patterns and remain with only relevant ones and then put them in the form that make them easily understood. The framework is validated basing on the experimental results that were obtained in Chapter Two and Three and will further be validated in Chapter Five.
CHAPTER FIVE

CRaPES: AN FP-GROWTH BASED PROTOTYPE FOR CRIME REPORTING AND PATTERNS EXTRACTION IN TANZANIA\(^6\)

This Chapter presents a prototype, named Crime Reporting and Pattern Extraction System (CRaPES) which was developed to implement the FP-Growth scaling method proposed in Chapter Three and partly validate the crime patterns mining framework proposed in Chapter Four. The prototype also enables the law enforcement agencies in Tanzania to report crimes and extract useful patterns from crime datasets. CRaPES is a desktop application developed by using Java. It consists of two modules; crime reporting and patterns extraction modules. After its development, a team of information systems development experts and end-users were invited to test and evaluate it. Evaluation results showed that the system is generally user friendly and is capable of not only reporting crimes but also extracting patterns.

5.1 Introduction

The use of a multiple minimum support approach basing on Shannon Entropy as an enhancement method for FP-Growth in the mining of crime patterns was proposed in Chapter Three of this dissertation. The Chapter puts forward that the use of a single minimum support approach is not appropriate for crime patterns mining. The reasons were twofold; first, crime datasets consist of items that vary greatly in terms of frequency of occurrence which makes a single minimum support approach suffers the rare item problem, and second, crime data steadily increase with time and thus a single minimum support approach becomes less effective with the growing amounts of data. The solution for this was to come up with an approach that automatically assigns minimum support values to each of the crime items in the dataset. This could be achieved by applying an aggregate function that studies the entire dataset and automatically assigns each crime item in the dataset a minimum support value basing on how frequently it has

\(^6\)This Chapter is based on the research paper: Matto, G. and Mwangoka, J. An FP-Growth based prototype for Crime Reporting and Pattern Extraction in Tanzania. (Manuscript)
appeared in the dataset. In connection, Chapter Four suggested a generic framework for extraction of crime patterns from multiple data sources. A framework requires step by step approach to have the patterns extracted from either structured or unstructured data.

This Chapter presents a developed working prototype that implements the solutions proposed in Chapter Three and Four. The aim was to evaluate effectiveness of the proposed FP-Growth scale-up method and partly implement and validate the proposed framework. Specific focus was to come up with a prototype that allows users to input crime dataset (either one crime by one, or import the dataset file) to the system, and then the system should apply the proposed FP-Growth scaling approach to obtain MIS values of each of the crime item in the dataset and then use them to extract useful patterns of crime from dataset. However, although in the Tanzania Police Force (TPF) website there is a form for the general public to report crimes (Fig. 21), this study observed that the practice of reporting and recording crimes in most of police stations in the country is based on the traditional manual systems which creates difficulties in management and processing of crime data. This observation is also reported in Robins (2009). According to the author, lack of effective record keeping of criminal records by the TPF has stifled criminal control mechanisms, and the slowness of criminal trials impacts negatively on police work.

![Figure 21: Web interface for public crime reporting](image-url)
Therefore, apart from developing a prototype that performs only extraction of patterns of crime from already collected datasets the study integrated a crime reporting module in the developed prototype. The module was aimed at improving reporting and management of crimes. We thus named this prototype *Crime Reporting and Pattern Extraction System (CRaPES)*.

5.2 Methodology

The development of this system followed a prototyping software development approach. As Dennis *et al.* (2008) explain, prototyping approach performs the analysis, design and implementation phases concurrently in order to quickly develop a simplified version of the proposed system and give it to the user for evaluation and feedback. The prototype is then improved following reactions and comments from the user. The improved prototype is given back again to the user for further evaluation, and the cycle continues until the user is satisfied with the final prototype.

Since this system was designed primarily to help TPF easily report and analyze crime data, and to implement and evaluate effectiveness of the proposed FP-Growth scaling method as well as crime patterns mining framework, police officers and information systems development experts were involved in the process of development of this prototype. Police officials from Intelligence, Data and Information and Communication Technology (ICT) sections of the TPF head office were interviewed during data collection. Appendix 1 shows questions that guided the interviews. Information systems development experts were those with knowledge in the area under study. These provided useful technical feedbacks that helped to improve the prototype.

As a result of data and system requirements that were gathered from users, the study came up with this system. As the name suggests CRaPES consists of two modules: Crime Reporting Module (which involves managing and reporting crimes), and Patterns Extraction Module (which involves crime patterns extraction). Crime Reporting and Pattern Extraction System (CRaPES) is a desktop application developed by using Java. The system can be accessed on a stand alone machine as well as on the shared environment via local or wide area networks.
Since CRaPES involves storing of crime data (i.e. crime reporting) there was a need to having a database of reported crimes. Thus the system was integrated with a separate crimes database (CRaPES database). SQLite was used to develop the database.

![Architectural Diagram of CRaPES](image)

**Figure 22: Architectural Diagram of CRaPES**

Figure 22 shows architectural diagram of this system. As the figure shows, CRaPES accepts input data from two sources. Its inbuilt database (which receives data from the Crime Reporting Module) and external sources. It should be noted that external data sources are involved but are optional.

### 5.3 CRaPES Implementation

Crime Reporting and Pattern Extraction System (CRaPES) was successfully developed and is up and running. Figure 23 shows CRaPES dashboard and subsequent sections describe more about the system.
Figure 23: CRaPES dashboard

5.3.1 Crime Reporting Module

This module helps law enforcement agencies to register reported crimes in the database. Datasets on the stored crimes are then analyzed to discover interesting patterns. These are actually the systems internal source of crime data. The crime reporting module consists of two main functional areas; Manage Crime and Report Crime.

(i) Manage Crimes Interface

As it was explained in Chapter One of this dissertation, what is regarded as crime in one domain may not necessarily be crime in another domain. So, this interface allows definition (or setting) of crimes as per the user’s domain. Crime Reporting and Pattern Extraction System (CRaPES) considers two parts in regard to crime: crime name, and crime category. All of these parts have to be defined by the user. Figure 24 is the Manage Crime interface. In this interface a list of some of crimes in Tanzania was defined, in consultation with the law enforcement agencies. The list can be added or reduced as per users needs. All crimes that will be added and appear in this crimes list will also appear in the Report Crime interface where user will only be
required to select the concerned crime(s) when recording reported crimes. Source codes for the implementation of this interface is shown in Appendix 7.

![CRiPES | Manage Crime](image)

**Figure 24:** Manage Crimes Interface

(ii) Report Crimes Interface

After crimes have been defined in the Manage Crime interface, in this interface user will record the actual crimes as they will be reported to police stations. Information that are to be recorded in this interface are basically the description of the crime (i.e. crime name, crime category, date in which crime was committed, location in which crime was committed) and the description of criminal (i.e. criminal gender, age, marital status, job, and residence). These are the 5WHs of crimes as explained in Chapter One.

As it have already been explained, the names and categories of crimes are defined in the Manage Crime interface (Fig. 24). In this Report Crimes interface user will only select crimes that have already been defined. The reason for this is to limit users self-defined crimes and to avoid multiple names that could mean the same crime. In fact, if this would be allowed to occur it would possibly lead to the extraction of erratic patterns. Figure 25 is the interface for crime reporting, and Appendix 8 shows source codes for the implementation of this interface.
Figure 25: Report Crimes Interface

Figure 26 shows a description of some of the crimes that were reported. As it can be seen in the figure, crimes are reported in their normal text forms, but the system puts them in text form as well as in their numerical equivalencies. The reason for adding numerical equivalencies is to allow the FP-Growth algorithm to easily analyze them.

So, when FP-Growth algorithm is to be used a file with crimes presented in numerical values will be exported to an external file, which will then be analyzed to discover patterns. This file can be exported by clicking on the File menu then Export Files as Fig. 27 shows.

The dataset can also be exported with crimes represented in normal texts (users may need to have them in normal text form for other uses than patterns extraction). This can be achieved by clicking on the File menu then Export Reported Crimes as shown in Fig. 28. This format will however not be executed by our patterns extraction prototype unless it is converted to numeral.
5.3.2 Pattern Extraction Module

This module allows user to extract patterns from crime datasets. FP-Growth algorithm is used to mine frequent itemsets and generate association rules. CRaPES includes both classical FP-Growth algorithm and the MIS based FP-Growth (i.e. the proposed Entropy based FP-Growth). The Entropy based FP-Growth is an enhancement of the classical FP-Growth that was proposed in Chapter Three. Two algorithms have been included in our prototype to provide user with freedom to use any of the two depending on the mining task.

(i) Classical FP-Growth Algorithm

This algorithms interface can be accessed by clicking on the Classical FP-Growth algorithm button on the dashboard or, alternatively, by clicking on the Extract Crime Patterns menu on the menu bar then Classical FP-Growth Algorithm. To generate crime patterns using Classical FP-Growth, as shown in Fig. 29, the crime dataset (crime dataset can be exported from CRaPES database) should be imported to the system, minimum support and confidence be set, and then the Run Algorithm button is clicked. The classical FP-Growth algorithm was, however, not the
basis for the development of this prototype.

(ii) Entropy Based MIS FP-Growth

This was one of the main reasons for the development of this prototype. The interface to this algorithm can be accessed by clicking on the *Entropy Based MIS FP-Growth* button (on the dashboard) or, alternatively, by clicking on *Extract Crime Patterns* menu on the menu bar then *Entropy Based MIS FP-Growth*. Figure 30 is the interface of the Entropy based FP-Growth algorithm. In order to run this algorithm, two files are required; ‘crime dataset’ and the ‘MIS file’. These files can be obtained from CRaPES database or imported from other external sources.

Files can be imported from other external sources so as to increase usability and flexibility of
the prototype. So, if there will be a need to extract patterns from crime dataset obtained from somewhere else, user will just have to import the two files to CRaPES and then click on the Run Algorithm button to get the patterns. If there is only crime dataset file without MIS file, CRaPES can be employed to calculate MIS values (basing on the proposed Entropy based method) and obtain the required MIS file.

The prototype provides two kinds of output, frequent itemsets and the system’s statistical summary (i.e. number of transactions in the dataset, maximum memory used, and the total run time). Figure 31 is part of the output showing some patterns generated from crime data stored in CRaPES database.

5.3.3 Exporting Crime Dataset and MIS File

Crime datasets and MIS files are the two input files for the MIS based FP-Growth algorithm. These files are exported from CRaPES database by clicking on the File menu on menu bar then Export Files (Fig. 27). Crime dataset can be obtained by clicking on Export Crimes Dataset button where a dialog box about the location in which the file is to be saved will appear.
Concerning MIS values, before exporting them user will be needed to calculate them first by clicking on ‘Calculate MIS Values’ button. This will list all reported crimes with their MIS values. Thereafter user will click on ‘Export MIS Values’ where he will select where to save the file with MIS values. Figure 32 shows some of the calculated MIS values from stored data.
Figure 31: CRaPES output showing some of the patterns generated

Figure 32: Some of the calculated MIS values
5.4 Results and Description of the Patterns Extracted

By using this prototype several frequent crimes were extracted from stored data. Figure 33 shows a list of such frequent crimes (together with their frequencies of occurrence) that were mined from CRaPES database. As seen in the figure, robbery of a person was committed most, followed by threatening to injury, and using of drugs. Concerning rare crimes, the following were extracted; conspiracy to kill, bicycle theft, theft in a bank, abduction/kidnapping, public fighting and unlawful sale of weapons. By the use of this prototype both frequently occurring crimes and those that occur rarely were extracted. But, the Least Support (LS) was set to be 3; so all items appeared in the dataset with a frequency of less than 3 were considered infrequent and thus ignored.

![Figure 33: Crimes extracted from CRaPES database](image)

Concerning association rules between the extracted crimes, several of them were generated. The strongest rule was between Threatening to injury and Robbery of a per-
son, where \( \{\text{Threatening to injury}\} \rightarrow \{\text{Robbery of a person}\}[\text{Supp} : 12] \). The next were \( \{\text{Robbery of a person}\} \rightarrow \{\text{Possession of weapon}\}[\text{Supp} : 8] \), and \( \{\text{Engaging in a riot}\} \rightarrow \{\text{Threatening to injury}\}[\text{Supp} : 7] \). Robbery of a person was also identified as the most occurring crime in the dataset (Fig. 33). This tells, therefore, that the law enforcement agencies in Tanzania (which data was used) should put appropriate measures against \( \{\text{Robbery of a person}\} \) as equally as to other crimes extracted from the data source. Figure 31 presents more of other rules generated. In this figure, the generated rules are visualized in an easy understandable textual description.

### 5.5 CRaPES Evaluation

In order to ensure effectiveness of the developed prototype both technical and usability evaluation were done. Evaluation criteria were therefore based on the systems user friendliness and on the systems ability to extract patterns. Systems development experts did technical evaluation on the user friendliness of the system while end-users did systems usability evaluation. Evaluation results are grouped into technical evaluation and end-user evaluation as further describes in the following sections.

#### 5.5.1 Technical Evaluation

A total of ten (10) information systems development experts were invited to undertake technical evaluation on this system. Specifically, they were asked to evaluate CRaPES on the following aspects: accessibility, easy to use, navigation, consistency, visual clarity, feedback, scalability and separation of concerns. Evaluators were asked to rate each of the presented aspects into a 5 points Likert Scale as either Very High, High, Average, Low or Very Low. Figure 34 summarizes results obtained from technical evaluation.

Three presented aspects (i.e. easy to use, navigation, visual clarity and scalability) were rated only Very High and High. This tells, therefore, that all of the evaluators were satisfied with these aspects. Accessibility and Consistency were rated Very High and High by most of evaluators.
but one evaluator on each of these aspects rated them Average. Moreover, Feedback was rated High by four evaluators, Average by five evaluators and Low by one evaluator. At the time this evaluation was done, there was no link on the system that allowed users to provide direct feedback via the system. Feedback could only be provided through email or phone numbers that were included in the system. The last aspect evaluated was Separation of Concerns. This aspect was rated Average by four evaluators, Low by five evaluators and Very Low by one evaluator. In fact, one of the main reason for developing CRaPES was to evaluate effectiveness of our proposed FP-Growth scaling method, thus when developing this system multiple users with different priority levels (separation of concerns) were not considered, the main focus was to see if the developed system is up and running and capable of not only reporting crimes but also employing our proposed FP-Growth scaling method to extract useful patterns of crime.

![Figure 34: Technical Evaluation Results](image)

5.5.2 End-User Evaluation

End-users are the ones involved in the day-to-day use of the system. The targeted users, for this evaluation were, therefore, the police officers. Consequently, five police officers were invited to test CRaPES and provide their evaluation feedback. Evaluation aspects in this case were on: ability of the system to define and redefine crimes as per the country, ability of the system to support reporting of crimes, ability to extract crime patterns, clarity of the patterns extracted, helpfulness or usefulness of the patterns extracted, and ability of the system to extract rare
5.6 Chapter Conclusion

This chapter presented a prototype, named Crime Reporting and Pattern Extraction System (CRaPES), that was developed based on three main reasons. First, to enable police and other the law enforcement agencies in Tanzania to register reported crimes and to extract useful patterns of crimes from stored datasets. The second reason for developing CRaPES was to evaluate effectiveness and applicability of the proposed FP-Growth scaling method. And a third reason for coming up with this prototype was to partly validate the framework for extraction of frequent patterns of crime from multiple sources of data as proposed in Chapter Four. System requirements for this prototype were obtained from the Tanzania Police Force, head office. The prototype was up and running and was capable of not only reporting crimes but also extracting useful patterns from crime data stored in CRaPES database or in any other external sources. Results on the evaluations that were performed by information systems development experts and end-users showed that the system is generally user friendly.
CHAPTER SIX

GENERAL DISCUSSION, CONCLUSION AND RECOMMENDATIONS

6.1 General Discussion

The spate of criminal threats in Tanzania, as in many other parts of the world, has been on the increase in the last few years. The successes recorded by criminals have been attributed by improper mechanisms by police and other law enforcement agencies that would enable preventive measures against criminals actions. In order to support the primary object of an efficient police, that is, prevention of crime; proactive measures are needed to preempt further crimes. Frequent Pattern Mining stand to aid in finding emerging patterns, series, and trends in the crime data. This will eventually help TPF and other law enforcement agencies to understand crime trends and predict or forecast future occurrences and thus improve preventive measures against crimes.

The growing volumes of texts and other kinds of data presents potentials for the extraction of useful frequent patterns of crime. Unfortunately, frequent pattern mining algorithms, FP-Growth in particular, do not scale up well with the growing volumes of data and it suffers from the rare item problem. It was on this ground that this research was carried out to propose ways in which FP-Growth algorithm can be improved for effective mining of frequent patterns of crime from growing volumes of data in Tanzania. The research was successfully carried out and the general discussion of the key findings are presented in subsequent sections.

6.1.1 Frequent Pattern Mining for Crime Detection

Frequent pattern mining can be an effective tool for helping police and other law enforcement agencies to improve strategies for crime prevention in the country. This was revealed in our study where, through an employment of frequent pattern mining and specifically the FP-Growth algorithm, our study was able to discover a number of crimes that commonly occur in the country and the magnitude of their occurrence. This result can be very helpful to the law enforcement agencies as it tells of which crimes are most commonly committed. The result can help them
to strategize on crimes that require more and urgent attention. The study’s results were based on the patterns that were extracted from news articles obtained from four selected daily published newspapers. A patterns mining model that was built on RapidMiner was employed in the patterns extraction.

The study showed (through the generated association rules) how the mined crimes were related. Specifically, the study observed a strong relationship between killing and other crimes such as brutality, gunned crimes, explosives, and traditional armed crimes. In fact, killing was found to be a consequent of each of these other crimes which were established to be antecedents to it. Therefore, according to this result, it was fair to say most of other crimes that were committed at the time these data were collected resulted into killings, and therefore if such crimes are contained, killings will also be contained.

Moreover, as it was described in the study the mined crimes were mapped per regions of the country in which they occurred. According to the results, crimes were reported in almost half of all regions of Tanzania where Mwanza region was leading by having the highest number of crimes reported in the period data was collected. Some few regions had moderate number of crimes reported, and nearly half of the region did not report any crime at the period in which data was collected. This finding can help the law enforcement agencies and other stakeholder including the general public to take necessary and reasonable preventive measures in the regions that report high crime rates.

6.1.2 FP-Growth Improvement Method

This study was specifically based on the FP-Growth algorithm for frequent patterns extraction. Unfortunately, this algorithm does not scale up well with the growing volumes of data. Besides, when the algorithm is to be used in crime datasets it suffers the rare item problem. The reasons are twofold. First, if the algorithms minimum support is set too low, huge amounts of crime patterns (especially with those crimes that are commonly committed) will be generated. On the other side, if it set too high lots of interesting pattern, including seasonal crimes patterns, will be lost.
To tackle the challenge, this study proposed the use of multiple minimum supports approach. The proposed approach was based on Shannon Entropy. Unlike other existing MIS approaches that suffer some challenges such as difficulties in stating "good" MIS value for each item and constantly tuning the minimum support values to obtain the best values, our proposed approach scans the entire dataset and automatically assigns MIS values on each of the crime item in the dataset basing on how frequently it has occurred in the dataset. In this way the proposed approach handles not only those crime items that rarely appear in the dataset but also those that appear very frequently. In other words, the proposed approach handles each crime item separately depending on how frequently it has appeared in the dataset.

### 6.1.3 Generic Framework for Mining Frequent Patterns of Crime

In the third specific objective of this study, a generic framework for the extraction of crime patterns from multiple data sources was proposed. The framework was proposed to confront the challenge that were observed in most of the existing similar frameworks. Some of those challenges are lack of input data flexibility, lack of clear procedures for cleaning and pre-processing the input data, as well as the rare item challenge in crime datasets. The proposed framework confronts these challenges in a four stages process. First stage is data collection. This stage allows flexibility for both structured and unstructured data to be employed in the mining process. Second stage is data cleaning and pre-processing. Depending on the type of input data, cleaning and preprocessing may involve replacing missing values, tokenizing, transforming cases, stemming, filtering stopwords and discretization. Pattern generation is the framework’s third stage. In this stage patterns are generated by using our proposed multiple minimum supports FP-Growth algorithm. In the final stage, pattern evaluation and visualization, the generated patterns are investigated to eliminate irrelevant patterns and remain with only relevant ones and then put them in the form that make them easily understood.
6.1.4 Crime Pattern Mining Prototype

A fourth objective of this study was to develop a working prototype for crime patterns extraction. This prototype, named Crime Reporting and Pattern Extraction System (CRaPES), was developed based on three main reasons. First, to enable police and other the law enforcement agencies in Tanzania to register reported crimes and to extract useful patterns of crimes from stored datasets. The second reason for developing CRaPES was to evaluate effectiveness and applicability of the proposed FP-Growth scaling method. And a third reason for coming up with this prototype was to partly validate the framework for extraction of frequent patterns of crime from multiple sources of data that was also proposed in this study. System requirements for the prototype were obtained from the TPF, head office. The prototype was up and running and was able to not only allow reporting of crimes but also extract useful patterns from both stored in CRaPES database and in other external sources. Results on the evaluations that were performed by information systems development experts and system users showed that the system is generally user friendly and is capable of not only reporting crimes but also extracting patterns.

6.2 General Conclusion

Frequent pattern mining is an effective tool for helping police and other law enforcement agencies to improve strategies for crime prevention in the country. However this study observed that TPF still relies on traditional and semi-automated methods to record and analyze crime data. As a result, some useful crime insights that could be discovered if frequent pattern mining could be employed might be missed out.

In connection, the study concurs with other existing literature on the effectiveness of FP-Growth algorithm in crime patterns mining. However, it was established that classical FP-Growth not only suffers the rare item problem but also does not scale up well with the growing volumes of data. This challenge is attributed mainly by the algorithms user-defined minimum support approach. In fact, a single user-defined minimum support is not appropriate for massive amounts of crime data. This study, therefore, proposed an FP-Growth scaling method that employs the
Shannon Entropy method to find the minimum supports of each of the crime item in the dataset. The proposed approach was tested on static crime data and established to be reasonably effective. The proposed approach was also evaluated through the developed prototype and found to be user friendly and capable of extracting useful crime patterns in the growing volumes of data.

In connection, the study concludes that a crime pattern mining framework that consider multiple sources of input data is more ideal for crime pattern mining.

6.3 General Recommendations

Following the findings from this study and the general conclusion that have been stated, the study puts forward the following general recommendations.

(i) TPF should embark on data (inside and outside their boundaries) for the fight against crime.

(ii) In addition to considering storing and managing its crime data electronically, TPF should employ frequent pattern mining as an effective way to extract useful crime patterns from available crime datasets.

(iii) For the purpose of this study, the crime pattern mining model extracted patterns from news articles that were collected in the period of two months only. It is recommend that other researches who will be interested in doing similar research should consider collecting same data in a longer period of time so as to observe if there is seasonality of patterns mined. It is equally recommended for involvement of more other data sources.

(iv) The proposed FP-Growth scaling method was tested and validated on static datasets. It is recommended other researchers to extend experimentations of the same method on streaming data.

(v) The study, furthermore, recommends that other researchers extend experimentations of the proposed methods on distributed environments such as Hadoop/MapReduce framework.
(vi) Since CRaPES was developed as a desktop application, the study recommends for further studies by other scholars to extend it into both web-based and mobile applications.

(vii) Although the proposed framework considered input data from structured and unstructured sources, validation of the framework was done on structured data only. It is therefore recommended for further validation of the framework on more other sources of crime data as described in the framework.

6.4 Future Works

6.4.1 More Datasets and More Data Sources

Although experimental results on crime patterns detection from news articles can create basis for improving strategies for crime prevention, such experiments were based on only one source of crime data (i.e. newspapers) in which data was also collected in a relatively short period of time (i.e. two months). There is a need therefore to extend this work to collect such data in a longer period of time to see if similar results will be obtained. In connection, future work should also involve more other sources of data from the media. The use of such different sources of data will also help to further validate the proposed framework.

6.4.2 Live, Streaming and Big Data Processing

Another area for future work is extension of experimentations on the proposed FP-Growth improvement method on data that is generated continuously by different sources such as mobile and web generated data as well as cloud data. This future work should also involve employment of distributed computing and Big Data processing.
6.4.3 Mobile and Web-based Applications for Crime Patterns Mining

Future work can also be carried out to develop a mobile and web-based system for crime pattern mining basing on the prototype that was developed in this study. In order for the developed extended system to be useful to the TPF and other law enforcement agencies, it should be capable of helping such law enforcement agencies to report crimes and extract patterns from stored crime data.

It is also important that the improved systems especially the mobile application to automatically identify the location of someone reporting crime and record that to the system. By so doing, crime reporting system will be more intelligent in plotting locations with high crimes for proper interventions.
REFERENCES


APPENDICES

Appendix 1: Interview Guide Questions

A: Informed Consent:

My name is George Matto, a PhD student at the Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha. I am undertaking a PhD research titled Effective Mining of Crime Patterns from Growing Volumes of Data Using Improved FP-Growth Algorithm. As part of data collection for this study I am requesting you to participate in an interview. Involvement in this interview is voluntary, so you may choose to participate or not. The interview will take approximately 40 minutes.

Your responses to interview will be used for academic purposes only and will be kept confidential. Please feel free to ask any question that you may have about this study.

B1: Police Intelligence Section:

1. How do you know the daily crime status/situation in the country?
2. How do you anticipate possible occurrences of crime?
3. How is crime prevention pursued? (What is your crime predictive measures?)
4. Do you have technological systems/tools to predict or anticipate crimes occurrence and prevent them? (Yes, No)
5. If your answer in 4 above is ‘Yes’?
   a. What are they?
   b. Do they involve the use of data?
6. Are online platforms (such as social media) support the increase of crimes in the country? (Yes, No)
7. If your answer in 6 above is ‘Yes’, how?
8. Have you employed online platforms (such as social media) to support the fight against crime?
9. If your answer in 8 above is ‘Yes’, how?
10. Do you make use of data in your crime prevention activities? (Yes, No)
11. If your answer in 10 above is ‘Yes’, what are those data? And, how are you using them?
B2: Data/Statistics Section:

1. What is the crimes trend in Tanzania in the last five years? (increasing, decreasing, constant)
2. If the trend in 1 above is increasing what are the possible causes for the increase? (unemployment, population growth, inadequate crime prevention strategies, etc.)
3. If the trend in 1 above is decreasing or constant what are the possible causes?
4. How do you keep/store records of crimes? (manual/paper files, computerized systems, etc.)
5. Do you analyze stored crime datasets? (Yes, No)
6. If your answer in 5 above is ‘Yes’;
   a. What is the aim of such analysis?
   b. How do you analyze your crime datasets?
7. Do you have stored records of unreported crimes? (Yes, No)
8. If your answer in 7 above is ‘Yes’;
   a. Do you analyze them?
   b. How?
9. What challenges are you facing in the management and analysis of crime data?

B3: Information and Communication Technology (ICT) Section:

1. Do you have computerized systems to support crime reporting at police stations? (Yes, No, Yes and No)
2. Do you have systems to allow the general public report crimes that happens in their localities? (Yes, No)
3. Do you keep records of reported crimes? (Yes, No)
4. If your answer in 3 above is ‘Yes’;
   a. For how long do you keep such records?
   b. How do you keep them? (Paper files, computerized systems, etc.)
5. Do you analyze stored crime datasets? (Yes, No)
6. If your answer in 5 above is ‘Yes’;
   a. What is the aim of such analysis?
   b. How do you analyze your crime datasets?
7. Do you have stored records of unreported crimes? (Yes, No)
8. If your answer in 7 above is ‘Yes’, do you analyze them? (Yes, No)
9. If your answer in 8 above is ‘Yes’, how do you analyze them?
10. What challenges are you facing in the management and analysis of crime data?
Appendix 2: Sample Newspaper Articles Used - Mtanzania Newspaper
Appendix 3: Sample Newspaper Articles Used - Nipashe Newspaper
Appendix 4: Sample Newspaper Articles Used - Mwananchi Newspaper
Appendix 5: Source Codes for MIS Tree Construction

// Source Codes that were adopted for MIS Tree Construction
// The tree is constructed based on the extracted MIS file
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MISTree {
    List<Integer> headerList = null;
    Map<Integer, MISNode> mapItemNodes = new HashMap<Integer, MISNode>);
    Map<Integer, MISNode> mapItemLastNode = new HashMap<Integer, MISNode>);
    MISNode root = new MISNode();

    MISTree() {
    }

    public void addTransaction(List<Integer> transaction) {
        MISNode currentNode = root;
        for (Integer item : transaction) {
            MISNode child = currentNode.getChildWithID(item);
            if (child == null) {
                MISNode newNode = new MISNode();
                newNode.itemID = item;
                newNode.parent = currentNode;
                currentNode.childs.add(newNode);
                currentNode = newNode;
                fixNodeLinks(item, newNode);
            } else {
                child.counter++;
                currentNode = child;
            }
        }
    }

    void addPrefixPath(List<MISNode> prefixPath, Map<Integer, Integer> mapSupportBeta, int minMIS) {
        int pathCount = prefixPath.get(0).counter;
        MISNode currentNode = root;
        for (int i = prefixPath.size() - 1; i >= 1; i--) {
            MISNode pathItem = prefixPath.get(i);
            if (mapSupportBeta.get(pathItem.itemID) < minMIS) {
                continue;
            }
            MISNode child = currentNode.getChildWithID(pathItem.itemID);
            if (child == null) {
                MISNode newNode = new MISNode();
                newNode.itemID = pathItem.itemID;
                newNode.parent = currentNode;
                newNode.counter = pathCount;
                currentNode.childs.add(newNode);
                currentNode = newNode;
            } else {
                child.counter++;
                currentNode = child;
            }
        }
    }
private void fixNodeLinks(Integer item, MISNode newNode) {
    MISNode lastNode = mapItemLastNode.get(item);
    if(lastNode != null) {
        lastNode.nodeLink = newNode;
    }
    mapItemLastNode.put(item, newNode);
    MISNode headernode = mapItemNodes.get(item);
    if(headernode == null){
        mapItemNodes.put(item, newNode);
    }
}

void createHeaderList(Comparator<Integer> itemComparator) {
    headerList = new ArrayList<Integer>(mapItemNodes.keySet());
    Collections.sort(headerList, itemComparator);
}

void deleteFromHeaderList(int item, Comparator<Integer> itemComparator) {
    int index = Collections.binarySearch(headerList, item, itemComparator);
    headerList.remove(index);
}

void MISPruning(int item) {
    MISNode headernode = mapItemNodes.get(item);
    while (headernode != null) {
        if (headernode.childs.isEmpty()) {
            headernode.parent.childs.remove(headernode);
        } else {
            headernode.parent.childs.remove(headernode);
            for (MISNode node : headernode.childs) {
                node.parent = headernode.parent;
            }
        }
        headernode = headernode.nodeLink;
    }
}

void MISMerge(MISNode treeRoot) {
    if (treeRoot == null)
        return;
    for(int i=0; i< treeRoot.childs.size(); i++){
        MISNode node1 = treeRoot.childs.get(i);
        for(int j=i+1; j< treeRoot.childs.size(); j++){
            MISNode node2 = treeRoot.childs.get(j);
            if (node2.itemID == node1.itemID) {
                node1.counter += node2.counter;
            }
        }
    }
}
node1.childs.addAll(node2.childs);

treeRoot.childs.remove(j);
j--;

MISNode headernode = mapItemNodes.get(node1.itemID);
if(headernode == node2){
    mapItemNodes.put(node2.itemID,
    node2.nodeLink);
}
else{
    while (headernode.nodeLink !=
    node2){
        headernode =
        headernode.nodeLink;
    }
    headernode.nodeLink =
    headernode.nodeLink.nodeLink;
}
}
}
for (MISNode node1 : treeRoot.childs){
    MISMerge(node1);
}

public void print(MISNode TRoot) {
    if (TRoot.itemID != -1)
        System.out.print(TRoot.itemID);
    System.out.print(' ');
    for (MISNode node : TRoot.childs) {
        print(node);
    }
}
Appendix 6: Source Codes for MIS Values Calculation

//Calculating MIS values from crime database using Entropy equation

private void Calculate_MIS_Values(){
    try{
        String sqls = "select crimes from crimes_reported";
        pst=conn.prepareStatement(sqls);
        rs=pst.executeQuery();
        while(rs.next()){ search += rs.getString("crimes"); }
        String[] split = search.split(" ");
        double ks = 1;
        Arrays.stream(split).collect(Collectors.groupingBy(s -> s))
            .forEach((k,v) -> crimes_reported += k+" " + Math.round(ks/((- (ks/(double)v.size())*(Math.log(ks/(double)v.size())))))+" ");
        String[] splits = crimes_reported.split(" ");
        for(int i=1; i<splits.length; i=i+2){
            if(splits[i].length()>7){ splits[i]="2"; }
            else if(Integer.valueOf(splits[i])>=2){
                splits[i]=splits[i];
            }else{ splits[i]="2"; }
            crimes_mis += splits[i-1]+" "+splits[i]+"\n";
        }
        //end of obtaining MIS values
        ItemSets.setText(crimes_mis);
        crimes_mis = ""; crimes_reported = "";
    } catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    finally{
        try{
            rs.close();
            pst.close();
        } catch(Exception e){
            
        }
    }
}
Appendix 7: Source Codes for Manage Crime

//Manage Crimes

import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Date;
import java.text.SimpleDateFormat;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import net.proteanit.sql.DbUtils;
public class ManageCrimes extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs =null;
    PreparedStatement pst=null;
    private static final DateFormat dateNow = new SimpleDateFormat("dd/MM/yyyy");
    private static final DateFormat timeNow = new SimpleDateFormat("HH:mm:ss");
    public ManageCrimes() {
        initComponents();
        setLocationRelativeTo(null);
        conn=javaconnect.ConnectDb();
        Select_Data();
    }
    public void dispose(){
        super.dispose();
    }
    public void close(){
        WindowEvent winClosingEvent = new WindowEvent(this,WindowEvent.WINDOW_CLOSING);
        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);
        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);
    }
    private void Select_Data(){
        try{
            String sql = "select * from crimes";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tableCrimes.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
        finally{
            try{
                rs.close();
                pst.close();
            }
            catch(Exception e){
            }
        }
    }
}
Save Crimes
private void saveCrimeActionPerformed(java.awt.event.ActionEvent evt) {
    Date date = new Date();
    try{
        String sql = "insert into crimes (crime_name, date_reg, time_reg) values(?,?,?)";
        pst=conn.prepareStatement(sql);
        pst.setString(1, dataCrime.getText());
        pst.setString(2, dateNow.format(date));
        pst.setString(3, timeNow.format(date));
        pst.execute();
        dataCrime.setText("");
        JOptionPane.showMessageDialog(null, "Saved Successfully");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    Select_Data();
}
private void dataCrimeActionPerformed(java.awt.event.ActionEvent evt) {
}
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    close();
}
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ReportCrime().setVisible(true);
}
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ManageCrimes().setVisible(true);
}
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new NormalBased().setVisible(true);
}
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new MisBased().setVisible(true);
}
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ExportFiles().setVisible(true);
}
**Appendix 8: Source Codes for Reporting Crimes**

`//Report Crimes`

```java
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Date;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import javax.swing.JOptionPane;
import javax.swing.
public class ReportCrime extends javax.swing.JFrame {
    Connection conn=null;
    ResultSet rs =null;
    PreparedStatement pst=null;
    private static final DateFormat dateNow = new
    SimpleDateFormat("dd/MM/yyyy");
    private static final DateFormat timeNow = new
    SimpleDateFormat("HH:mm:ss");
    private String crimesReported = ""
    public ReportCrime() {
        initComponents();
        this.setLocationRelativeTo(null);
        conn=javaconnect.ConnectDb();
        Select_Data();
    }
    public void dispose(){
       super.dispose();
    }
    public void close(){
        WindowEvent winClosingEvent = new
        WindowEvent(this,WindowEvent.WINDOW_CLOSING);
        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);
        private void Select_Data(){
            try{
            String sql = "select * from crimes_reported";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tableCrimesReported.setModel(DbUtils.resultSetToTableModel(rs));
            }
            catch(Exception e){
                JOptionPane.showMessageDialog(null, e

    finally{
        try{
        }
    }
}
```
rs.close();
pst.close();
}
catch(Exception e){
}
}

/**
//Codes for initializing components have been skipped here
**/

private void runAlgorithm1ActionPerformed(java.awt.event.ActionEvent evt) {
    Date date = new Date();
    if(this.jCheckBox1.isSelected()){
        crimesReported += "1 ";
    }else{}
    if(this.jCheckBox2.isSelected()){
        crimesReported += "2 ";
    }else{}
    if(this.jCheckBox3.isSelected()){
        crimesReported += "3 ";
    }else{}
    if(this.jCheckBox4.isSelected()){
        crimesReported += "4 ";
    }else{}
    if(this.jCheckBox5.isSelected()){
        crimesReported += "5 ";
    }else{}
    try{
        String sql = "insert into crimes_reported (crimes,date_happened,location,claimer,suspect,date_reg,time_reg) values(?,?,?,?,?,?,?)";
        pst=conn.prepareStatement(sql);
        pst.setString(1, crimesReported);
        pst.setString(2, dateHappened.getText());
        pst.setString(3, location.getText());
        pst.setString(4, claimer.getText());
        pst.setString(5, suspect.getText());
        pst.setString(6, dateNow.format(date));
        pst.setString(7, timeNow.format(date));
        pst.execute();
        dateHappened.setText(""");
        location.setText(""");
        claimer.setText(""");
        suspect.setText(""");
        crimesReported = "";
        jCheckBox1.setSelected(false);
        jCheckBox2.setSelected(false);
        jCheckBox3.setSelected(false);
        jCheckBox4.setSelected(false);
        jCheckBox5.setSelected(false);
        JOptionPane.showMessageDialog(null, "Reported and Saved Successfully");
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
} 
Select_Data(); 
private void claimerActionPerformed(java.awt.event.ActionEvent evt) {
} 
private void locationActionPerformed(java.awt.event.ActionEvent evt) {
} 
private void dateHappenedActionPerformed(java.awt.event.ActionEvent evt) {
} 
private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {
} 
private void suspectActionPerformed(java.awt.event.ActionEvent evt) {
} 
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    close();
} 
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ReportCrime().setVisible(true);
} 
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ManageCrimes().setVisible(true);
} 
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new MisBased().setVisible(true);
} 
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new NormalBased().setVisible(true);
} 
private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new ExportFiles().setVisible(true);
Detecting Crime Patterns from Swahili Newspapers using Text Mining

George Matto* and Joseph Mwangoka

School of Computational and Communication Science and Engineering, The Nelson Mandela African Institution of Science and Technology, P. O. Box 447, Arusha, Tanzania.
Email: mattog@nm-aist.ac.tz
Email: josephwm@gmail.com
*Corresponding author

Abstract: The Tanzania Police Force, as many other law enforcement agencies in developing countries, relies mostly on manual, personal judgments, and other inadequate tools for analysis of data in its crime databases. This approach is inadequate and prone to errors. Moreover, research shows that more than half of all crimes committed in Tanzania are not reported to police and thus it is likely that they are not analysed by the police. In this study, we use text mining to extract crime patterns from sources of crime data outside police databases. In fact, we use four daily published Swahili newspapers. With the help of our developed patterns mining model we extracted several crimes reported in the newspapers, we mapped the distribution of the mined crimes country-wide, and with the use of FP-Growth, we generated association rules between the mined crimes. Results from this study will contribute to crime detection and prevention strategies.

Keywords: Crime, Crime Patterns, Text Mining, Association Rules, FP-Growth.

Reference to this paper should be made as follows: Matto, G. and Mwangoka, J. (2017) ‘Detecting Crime Patterns from Swahili Newspapers using Text Mining’, Int. J. of Knowledge Engineering and Data Mining.

Biographical notes: George Matto is a PhD scholar at the School of Computational and Communication Sciences and Engineering at the Nelson Mandela Institution of Science and Technology. He holds M.Sc. in Computer Science and B.Sc. (Hons) in Computer Science. His areas of research include Database Management Systems, Data Mining, Text Mining, Pattern Recognition and Big Data.

Dr. Joseph W. Mwangoka is a Senior Lecturer at the Nelson Mandela African Institution of Science and Technology, Arusha, Tanzania. He received his Ph.D. degree from Tsinghua University, Beijing, China in 2009. Until 2012 he was a Senior Research Engineer at the Institute of Telecommunications, Aveiro, Portugal. His research interests include Data Science, cognitive radio technology, dynamic spectrum management, ICT4D/E, health informatics, and cloud computing. He has co-authored a number of peer-reviewed book chapters, journal articles and conference proceedings.
1 Introduction

In Tanzania, reports from the Tanzania Police Force and the National Bureau of Statistics show a slight decrease of crimes reported in police stations. For instance, as Figure 1 (a) shows, the number of criminal offences reported in 2015 was 519,203, compared with 528,575 that was reported in 2014, a decrease which was equivalent to 1.8%. Similarly, there was a 6% and 1.1% decrease of crimes reported in 2014 and 2013 respectively (The United Republic of Tanzania, 2016; 2015; 2014; 2013). While such reports record a crime decrease, several researches indicate a continued rise of crime fear among Tanzanians. Crime fear is defined by Hale (1996) as the fear of being a victim of crime. In 2011, for example, 42% of Tanzanians were living with fear of becoming victims of crimes (Wambura, 2015a), 41% in 2012 (Gaddis et al., 2013), and over 45% in 2014 (Twaweza, 2014) as shown more in Figure 1 (b).

Reports about reduced number of reported crimes on one hand, and those of increased crime fear on the other hand presents a contradicting fact about the actual crime situation in the country. However, Jackson (2009) argues that there exists a link between fear of crime and likelihood of victimization, and that, high crime fearing rate is a natural response to crime incidents as it is grounded on the reality of crime. This is why Twaweza (2014) pointed out that the increased crime fear in Tanzania is a result of the increased crime incidents. In connection, research shows a low tendency of crime reporting in Tanzania. In 2011 to 2013 for example, 54 percent of people who were victims of crime did not report the incidents to police (Wambura, 2015b). Therefore, although police records show a decreasing rate of reported crimes, the incidents are likely on the increase.

Robust proactive measures are needed to support the prevention of further criminal incidents. In fact, this is the primary objective of an efficient police (Zaman, 2013). The advancements in science and technology plays a major role on this. Technologies can help in analyzing crime datasets to find emerging patterns, series, and trends. This will help police force to understand the current crime trends and predict or forecast future occurrences (Usher and Rameshkumar, 2014).

Unfortunately, Tanzania Police Force, as many other law enforcement agencies in developing countries, rely mostly on manual or personal judgments and other inadequate tools for inspection, exploration and analysis of crime data. Moreover, as said by Isafiade.
Detecting Crime Patterns from Swahili Newspapers using Text Mining

and Bagula (2013), the volume of data that can be processed simultaneously within a reasonable time frame is limited thus results into omission of complex and crucial relationships between different crimes attributes. Apart from the challenge of maximally exploring crime datasets, the Tanzania Police Force is faced with another challenge of capturing and analysing unreported crimes.

Newspapers, social media, and other similar platforms can be a useful source of crime data that are not necessarily reported in police stations. Text mining and other data mining techniques are capable of providing police with useful insights from such platforms. It is on this same line that this research was carried out to employ text mining to analyse and extract crime patterns from Tanzania’s Swahili newspapers. Swahili is a Tanzanian official language. Specifically, our objectives were three fold. To mine frequently reported crimes, to investigate on the distribution of crimes per regions, and to generate association rules between the mined crimes. Results from this study will be helpful to police and other law enforcement agencies in the process of crime detection and prevention.

The remainder of this paper is organized as follows: section 2 provides related studies; section 3 presents the methodology used; section 4 presents results and discussions; section 5 gives the conclusion and puts forward the future work.

2 Related Studies

Data mining is a process of extracting knowledge from huge amount of data stored in databases, data warehouses and data repositories (Jani, 2014). It can be achieved by Association, Classification, Clustering, Predictions, Sequential Patterns, and Similar Time Sequences (Hipp et al., 2002). In Association, the relationship of a particular item in a data transaction on other items in the same transaction is used to predict patterns. The idea of mining association rules originated from the analysis of market-basket data where rules like “if a customer buy bread he is 85% likely to purchase butter also” are generated. Today the generation of association rules is one of the most popular data mining methods. Moreover, association rules are not restricted to dependency analysis in the context of retail applications but are successfully applicable to a wide range of business problems.

Text mining refers to using data mining techniques for discovering useful patterns from texts. Data mining and text mining have become a powerful technique with great potential to help criminal investigators focus on the most important information on crime datasets, as such they help police investigating officers to identify hidden patterns from crime data (Varghese, 2010). A great deal of scientific research have consequently been performed on crime patterns mining (Gangavane and Nikose, 2015).

Several of such research, however, have been concentrated on identifying crime patterns from crime databases and other structured data. For example, Zubi and Mahmmud (2014) proposed model for crime and criminal data analysis using data mining techniques. They used Libyan national criminal record data for their experiment which was based on association rule mining and clustering. Another research by Isafiade and Bagula (2013) which focused on creating a flexible and effective solution to crime situation recognition used crime incident reports that were stored on various crime databases. Wang et al. (2012) proposed a pattern detection algorithm called Series Finder
that grows a pattern of discovered crimes from within a database, starting from a “seed” of a few crimes. Series Finder used data collected by the Crime Analysis Unit of the Cambridge Police Department to detect patterns of crime committed by the same individual(s).

Same trend of generating crime patterns from structured crime data is observed in Jani (2014) and Varghese et al. (2010). Elyezjy and Elhaless (2015) attempted to investigate crime patterns using text mining and network analysis by mining offenders’ names from unstructured text data in the Arabic language. However, they again did the mining from investigations documents that were obtained from police department. These, and other similar literature, show that there is a little research in methods and techniques that extract crime patterns from unstructured texts in other sources outside police crime datasets. There is similarly little research that consider the same extractions from datasets in local languages.

3 Methodology

3.1 Source of Data

In this study we used datasets obtained from four reputable Swahili newspapers. Selected newspapers were Majira, Mtanzania, Mwananchi and Nipashe. At least eight articles from each of the newspaper were collected and analysed. The articles were those with crime related news reports published in May, 2016. Swahili newspapers were used because of two reasons. First, most of the newspapers in Tanzania are published in Swahili and second, the selected newspapers had news reporters from all over the country and hence countrywide coverage.

3.2 Workflow of the Crimes Mining Process

The first step in the process of crimes mining was the collection of articles. The collected articles were then pre-processed before loaded to the crimes mining model that was built on RapidMiner Studio. Pre-processing involved moulding the articles obtained from various newspapers platforms into a suitable format.

The next step was ‘TEXT PROCESSING’. This is actually what we trained our text mining model to do. To accomplish this, it reads and process document. In the ‘read document’, the model loads pre-processed newspaper articles in .txt format. Several articles (of the same newspaper) were then combined together as one document and taken to the ‘process document’ step.

We trained the model to do four things in the ‘process document’ step. First was ‘tokenize’ in which words in the articles were grouped together and counted. Second was ‘filter stopwords’. Stopwords are the most common words in a language. For example, stopwords in the English language are such as the, is, at and which. RapidMiner has built-in dictionaries in several languages to find and filter stopwords out. Unfortunately, the articles that were used in this work were in Swahili, the language which dictionary is not available in RapidMiner. Third step was ‘transform cases’. Since RapidMiner is case sensitive where letters that are uppercase do not match with the same letters in lowercase, we opted to use lower cases. So, in this step, all letters were transformed into lower case. The fourth and last step was ‘replace tokens’. Tokens are words, phrases, symbols or other meaningful elements in the articles. Several of such tokens can be presented
Detecting Crime Patterns from Swahili Newspapers using Text Mining
differently but meaning the same thing. In this step, similar tokens were replaced by more
common ones, and their total occurrences were recorded. In this way, we were able to
obtain various reported crimes, their frequency of occurrences, and the regions in which
they occurred. This mining process workflow is summarized in Figure 2.

**Figure 2.** Workflow of the Mining Process

3.3 Association Rules Generation

3.3.1 Data Preparations for Rules Generation

After obtaining a set of mined crimes, we employed FP-Growth algorithm to generate
association rules between those crimes. Since the mining process was done from four
different newspapers, frequency of occurrences of the mined crimes varied from one
newspaper to another. To make FP-Growth applicable in our case, we organized the
mined dataset in tabular form in such a way that mined crimes were set as attributes of
the table and the newspapers where the crimes were mined were set as rows, as shown in
Figure 3. A Boolean value ‘TRUE’ or ‘FALSE’ was assigned in each attribute to indicate
whether a particular crime was reported in a particular newspaper or not.
3.3.2 Rules Generation Process

The prepared dataset was then loaded to RapidMiner for association rules generation. To accomplish rules generation we used ‘FP-Growth’ and ‘Generate Association Rules’ operators. But due to the nature of our dataset the FP-Growth could not be applied directly because it requires all attributes to be binominal. We then did some pre-processing to mould our dataset into the desired form. In fact, after retrieving the prepared crimes dataset, we used ‘Text to Nominal’ and ‘Nominal to Binominal’ operators to pre-process the data before FP-Growth was used. This process is shown in Figure 4.

4 Results

4.1 Mined Crimes and their frequencies of occurrence

Ten (10) crime incident types were extracted from the input files. Since the input files were in Swahili, the mined results were also in Swahili. The following are the results...
Detecting Crime Patterns from Swahili Newspapers using Text Mining

with their English translation in parentheses; Mauaji (Killings), Unyama (Brutality), Milipuko (Explosives), Makosa ya kingono (Sexual offenses), Ujambazi/Uvamizi (Invading/Gangs), Uhalifu wa kutumia Bunduki (Gunned crimes), Uhalifu wa kutumia Silaha za jadi (Traditional armed crimes), Ugaidi (Terrorism), Madawa ya kulevya (Drugs), and Mauaji ya Albino (Killing of people with albinism). Table 1 summarizes this finding.

<table>
<thead>
<tr>
<th>Types of Crimes as mined from newspapers</th>
<th>Majira Total Freq.</th>
<th>Majira No. of articles</th>
<th>Mtananzia Total Freq.</th>
<th>Mtananzia No. of articles</th>
<th>Mwananchi Total Freq.</th>
<th>Mwananchi No. of articles</th>
<th>Nipashe Total Freq.</th>
<th>Nipashe No. of articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mauaji</td>
<td>60</td>
<td>4</td>
<td>59</td>
<td>7</td>
<td>98</td>
<td>8</td>
<td>36</td>
<td>6</td>
</tr>
<tr>
<td>Unyama</td>
<td>17</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>17</td>
<td>6</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>Milipuko</td>
<td>15</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Makosa ya kingono</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>Ujambazi/Uvamizi</td>
<td>15</td>
<td>4</td>
<td>35</td>
<td>8</td>
<td>21</td>
<td>5</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>Uhalifu wa kutumia Bunduki</td>
<td>11</td>
<td>2</td>
<td>27</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Uhalifu wa Silaha za jadi</td>
<td>10</td>
<td>3</td>
<td>12</td>
<td>5</td>
<td>18</td>
<td>5</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>Ugaidi</td>
<td>2</td>
<td>1</td>
<td>45</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Madawa ya kulevya</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Mauaji ya Albino</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

As shown in the presented results, Mauaji (Killings) occurred mostly in all of the four newspapers. Its frequency of occurrence was 60 in the 4 articles of the Majira newspaper, 59 in 7 articles of the Mtanzania, 98 in 8 articles of the Mwananchi, and 36 in 6 articles of the Nipashe. It can further be observed that other crime incident types with high frequency of occurrence were; Unyama (Brutality), Milipuko (Explosives) and Ujambazi/Uvamizi (Invading/Gangs).

In the other hand, Mauaji ya Albino (Killing of people with albinism) were seldom reported. It appeared only in the Mwananchi newspaper, which tells that in May, 2016 there were very few incidents related to killings of people with albinism in Tanzania. The next least reported criminal incident was Madawa ya kulevya (Drugs), which was not reported at all in the Majira newspaper.

4.2 Crimes occurrence by Regions

Investigating crimes distribution per regions was a second objective of this study. Results showed that out of 30 regions of Tanzania, newspapers reported crime occurrences in 16 regions. Zanzibar islands consist of five regions, but for the purpose of this study we treat
the islands as a single region, that is, the region of Zanzibar. Figure 5 shows how crimes were distributed across the regions of Tanzania.

**Figure 5.** Map of Tanzania showing crimes distribution by region, in May, 2016

As shown in Figure 5, Mwanza region was leading by having an average criminal incidents of more than 10. Tanga followed with an average of 6 to 10 criminal incidents. Following these results, the two regions were categorized as relatively high crime zones in May, 2016. Consequently, the Tanzania Police Force and other stakeholders including the general public could consider placing extra efforts to reduce the high crime rates in the regions.

14 regions; four in the lake zone (Kagera, Geita, Shinyanga and Mara), two northern regions (Arusha and Kilimanjaro), two coast regions (Dar es Salaam and Pwani), two central regions (Dodoma and Morogoro), and three southern highlands regions (Rukwa, Mbeya and Njombe) and Zanzibar had an average of one to five criminal incidents. This made these regions to be categorized as low crime zones.

The four newspapers that were used in this study did not report any crime incident in southern regions (Ruvuma, Lindi and Mtwara), western regions (Kigoma, Katavi and Tabora) as well as Simiyu, Singida, Manyara and Iringa. Basing on this finding it can be fair to say that those regions were relatively safe in in the time the news were reported.

**4.3 Association Rules Generated**

Six association rules were generated among the mined crimes. Figure 6 is a graphical visualization of the generated rules while Figure 7 shows textual description of the rules.
Detecting Crime Patterns from Swahili Newspapers using Text Mining

**Figure 6.** Graphical visualization of the generated association rules

![Graphical visualization of the generated association rules]

**Figure 7.** Textual Description of the Generated Rules

- [Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)
- [Uhalifu wa kutumia Bunduki, Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)
- [Ugaidi, Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)
- [Milipuko, Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)
- [Uhalifu kwa silaha za jadi, Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)
- [Uhalifu kwa silaha za jadi, Unyama, Milipuko, Unyama] \(\rightarrow\) [Mauaji] (confidence: 1.000)

What can be inferred from the generated rules is that, since all the generated association rules concluded to *Mauaji* (Killings) then the premises (i.e. *Unyama* (Brutality), *Uhalifu wa kutumia Bunduki* (Gunned crimes), *Milipuko* (Explosives) and *Uhalifu kwa silaha za jadi* (Traditional armed crimes)) possibly resulted into killings. Therefore, it can be fair to say; if brutality, gunned crimes, explosives and traditional armed crimes are contained, killings will also be contained.

Considering the patterns that we have mined it is not surprising to see what have been reported by other researchers about the rise of crime fear. In fact, this research confirms the existence of crime incidents which are the contributing factors to the reported fear. Police, and other law enforcement agencies, can use these mined patterns to boost their crime detection and prevention strategies.

5 Conclusion and Future Work

The main contribution of this work is the extraction of crime patterns from unstructured data in Swahili newspapers. By applying data mining techniques we were able to analyse and mine crime patterns and then generate association rules between the mined crimes. We were also able to show distribution of crime occurrences per regions of Tanzania. Mined patterns can help police officers and other law enforcement agencies to understand the crime situation from a different angle, and thus put in place more efficient proactive measures against future crimes.
Future work is to collect data for longer period of time, mine and check if there will be seasonality of patterns. Also extract crime patterns from structured data in crime databases and establish correlation between patterns of crimes from what is being reported in police and what is reported in the media.

References


Detecting Crime Patterns from Swahili Newspapers using Text Mining

Wambura, P. M. (2015b) ‘Police corruption in Africa undermines trust, but support for law enforcement remains strong’. Afrobarometer Dispatch No. 56.


Mining Frequent Patterns of Crime Using FP-Growth with Multiple Minimum Supports based on Shannon Entropy

George Matto  
School of Computational and Communication Sciences and Engineering, Nelson Mandela African Institution of Science and Technology  
Arusha, Tanzania

Joseph Mwangoka  
School of Computational and Communication Sciences and Engineering, Nelson Mandela African Institution of Science and Technology  
Arusha, Tanzania

ABSTRACT
FP-Growth is one of the most effective and widely used association rules mining algorithm for discovering interesting relations among items in large datasets. Unfortunately, classical FP-Growth mines frequent patterns by using single user-defined minimum support threshold. This is not adequate for real-life applications, such as crime patterns mining. On one side, if minimum support is set too low, huge amount of crime patterns (including uninteresting patterns) may be generated, and on the other side, if it is set too high, lots of interesting patterns (including seasonal patterns) may be lost. This paper proposes the use of Multiple Item Support (MIS) thresholds instead of single minimum support to tackle the challenge. We employ Shannon entropy method to develop an algorithm that obtains MIS values from crime datasets. The proposed approach is tested on different sizes of input data via a developed working prototype. Experimental results show that our suggested approach outperforms classical FP-Growth in terms of running time and memory use.

Keywords
FP-Growth; Crime Pattern; Multiple Minimum Support; Shannon Entropy.

1. INTRODUCTION
Mining association rules is one of the key data mining tasks. It discovers interesting relations amongst items in large databases. An association rule, according to Jiawei et al. [1], is an implication of the form A → B, where, A is the antecedent while B is the consequent. It gives information of the form: when A appears then B will possibly appear too. The idea of discovering association rules begun from the investigation of market-basket data. It is on such investigations that rules like “if a customer buy bread he is 85% likely to purchase butter also” are generated. Today, association rule mining has become a powerful technique with huge potential and wide applications in several domains. One of such domains is crime patterns analysis, in which crime analysts mine association rules from crime datasets. Such rules help to discover patterns in criminal behaviour that can help to predict crime, anticipate criminal activities, and prevent further crimes [2, 3, 26].

There are several association rules mining algorithms. According to Kumbhare and Chobe [4], Apriori, FP-Growth and Eclat are the most widely used. Comparative studies among such algorithms (see for example [4] and [5]) indicate FP-Growth more efficient in terms of number of database scans, execution time and memory consumption. FP-Growth has also been used intensively in crime patterns mining [5, 6, 26].

Classical FP-Growth mines frequent patterns by using a single user-specified minimum support (abbreviated as minsup) [7]. However, using single minimum support for crime patterns mining is not adequate since it does not reflect the nature of each crime item in the dataset. If, for instance, such a minimum support is set too low, huge amount of crime patterns (including uninteresting patterns) will be generated and thus provides misleading results. On the other side, if it is set too high, many interesting patterns may be lost since some of crimes (e.g. killing of people with albinism in some countries like Tanzania) occur seasonally and thus rarely found in the dataset.

To tackle the challenge, in this paper we propose a method that replaces the minimum support value defined by user with an aggregate function that computes multiple minimum supports basing on empirical analysis of the dataset. Our proposed function is based on the Shannon entropy equation. We test our solution on four clusters of training data with different sizes and compare the run time and memory utilization of our algorithm versus classical FP-Growth. Experimental results revealed that our suggested solution is more effective in terms of run time and memory consumption.

The remainder of this paper is organized as follows; in Section 2 we present a survey of related literature. In this section we discuss various association rule-mining concepts, revisit the FP-Growth algorithm and review previous studies that attempted to improve the FP-Growth. In Section 3 we describe our proposed approach. In Section 4 we present our experimental results, and in Section 5 we conclude the paper.

2. RELATED LITERATURE
2.1 Association Rules Mining
Association rule mining problem is stated by [8] and [9] like this. Suppose I = {1, 2, 3, ..., n} and T are sets of finite items and transactions respectively. Suppose DB is a transaction database that comprises set of items in I. DB = T1, T2, T3, ..., Tn where Ti(i ∈ [1 ... n]). Suppose X and Y are itemsets, an itemset is a set of items in I i.e. X ⊆ I and Y ⊆ I. An association rule of itemsets X and Y is denoted as X → Y. This is the relationship between X and Y, given that X and Y are disjoint itemsets.

Support of an itemset X, represented as sup(X), is the fraction of transactions T, comprising X. Similarly, support of itemset Y, represented as sup(Y), is the fraction of transactions T, comprising Y.
\[
\text{sup}(X) = \frac{\text{Number of Transactions comprising } X}{\text{Total Number of Transactions}}
\]

\[
\text{sup}(Y) = \frac{\text{Number of Transactions comprising } Y}{\text{Total Number of Transactions}}
\]

Support of an association rule \(X \rightarrow Y\), denoted as \(\text{sup}(X \rightarrow Y)\), is the fraction of transactions comprising of both \(X\) and \(Y\).

\[
\text{sup}(X \rightarrow Y) = \frac{\text{Number of Trans' comprising } X, Y}{\text{Total Number of Transactions}}
\]

Confidence of an association rule \(X \rightarrow Y\), denoted as \(\text{conf}(X \rightarrow Y)\), is an indication of how often the rule have been found to be true. It is given as

\[
\text{conf}(X \rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}
\]

Zeng et al. [10] pointed out that for a given user-specified minimum support, \(\text{minsup}\), if the itemset meets the condition \(\text{sup}(X) \geq \text{minsup}\), then itemset \(X\) is regarded as frequent itemset and conversely itemset \(X\) is regarded as infrequent itemset.

### 2.2 FP-Growth Algorithm

FP-Growth is one of the most efficient association rule mining algorithms. The algorithm mines frequent itemsets without generating the candidates. FP-Growth algorithm was proposed by Han et al. [11] to overcome the multiple database scans and candidate generations of the Apriori algorithm [12]. According to Han et al. [11] FP-Growth uses a divide-and-conquer strategy to mine frequent itemsets. It uses two steps. First, build an FP-Tree by condensing a transaction database into a compressed structure. And second, extract itemsets directly from the FP-Tree that was built in the first step.

#### Step 1: Building an FP-Tree

According to Han et al. [11], the algorithm for FP-Tree construction requires two sorts of inputs; the transaction database (DB) or the dataset and the minimum support threshold. The general steps for the FP-Tree construction are as shown in algorithm 1.

**Table 1. FP-Tree construction**

<table>
<thead>
<tr>
<th>Algorithm 1: FP-Tree Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> DB, minsup ((\xi))</td>
</tr>
<tr>
<td><strong>Output:</strong> FP-tree</td>
</tr>
<tr>
<td><strong>Process:</strong></td>
</tr>
<tr>
<td>1. Scan DB once. Discard infrequent items and collect the set of frequent items, (F), (and their supports). Sort (F) in support-descending order</td>
</tr>
<tr>
<td>2. Scan DB again to construct FP-Tree.</td>
</tr>
</tbody>
</table>

We describe how this algorithm works by considering a transaction database, DB, in Table 2. DB has eight different crime items, i.e. crime1, crime2, crime3, crime4, crime5, crime6, crime7 and crime8 represented as \(C_1\), \(C_2\), \(C_3\), \(C_4\), \(C_5\), \(C_6\), \(C_7\) and \(C_8\) respectively. We consider \(\text{minsup}\) threshold for this case to be 2 (i.e. \(\xi = 2\)).

According to Algorithm 1 first, a scan of DB collects \(F\), the set of frequent items and the support of each of those frequent items, \(F = \{C_1:12, C_2:9, C_3:9, C_4:7, C_5:4, C_6:3, C_7:2\}\). Then sort \(F\) in support-descending order as FList, while discarding those items whose support is less than \(\text{minsup}\) threshold. FList is the list of frequent items. Since \(C_6\), \(C_7\) and \(C_8\) has support less than 3 they will be discarded and thus FList = \{\(C_1:12, C_2:9, C_3:9, C_4:7, C_5:4, C_6:3, C_7:2\}\). Second, we use this order, and the methods indicated in the FP-Tree construction algorithm to build an FP-Tree. Figure 1 is the constructed FP-Tree. Link is added to speed lookup and easy matching of the pointer to FP-Tree.

**Table 2. Transaction Database (DB)**

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T001</td>
<td>(C_1, C_2)</td>
</tr>
<tr>
<td>T002</td>
<td>(C_1, C_5, C_6)</td>
</tr>
<tr>
<td>T003</td>
<td>(C_3, C_4)</td>
</tr>
<tr>
<td>T004</td>
<td>(C_1, C_2, C_8)</td>
</tr>
<tr>
<td>T005</td>
<td>(C_3, C_4)</td>
</tr>
<tr>
<td>T006</td>
<td>(C_1, C_3)</td>
</tr>
<tr>
<td>T007</td>
<td>(C_1, C_2)</td>
</tr>
<tr>
<td>T008</td>
<td>(C_5, C_6)</td>
</tr>
<tr>
<td>T009</td>
<td>(C_3, C_4, C_7)</td>
</tr>
<tr>
<td>T010</td>
<td>(C_1, C_2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T011</td>
<td>(C_1, C_2)</td>
</tr>
<tr>
<td>T012</td>
<td>(C_1, C_3)</td>
</tr>
<tr>
<td>T013</td>
<td>(C_1, C_2)</td>
</tr>
<tr>
<td>T014</td>
<td>(C_2, C_5, C_6, C_7)</td>
</tr>
<tr>
<td>T015</td>
<td>(C_3, C_4)</td>
</tr>
<tr>
<td>T016</td>
<td>(C_1, C_2, C_4)</td>
</tr>
<tr>
<td>T017</td>
<td>(C_3, C_4)</td>
</tr>
<tr>
<td>T018</td>
<td>(C_1, C_3)</td>
</tr>
<tr>
<td>T019</td>
<td>(C_1, C_2, C_5)</td>
</tr>
<tr>
<td>T020</td>
<td>(C_3, C_4)</td>
</tr>
</tbody>
</table>

According to Algorithm 1, a scan of DB collects \(F\), the set of frequent items and the support of each of those frequent items, \(F = \{C_1:12, C_2:9, C_3:9, C_4:7, C_5:4, C_6:3, C_7:2\}\). Then sort \(F\) in support-descending order as FList, while discarding those items whose support is less than \(\text{minsup}\) threshold. FList is the list of frequent items. Since \(C_6\), \(C_7\) and \(C_8\) has support less than 3 they will be discarded and thus FList = \{\(C_1:12, C_2:9, C_3:9, C_4:7, C_5:4, C_6:3, C_7:2\}\). Second, we use this order, and the methods indicated in the FP-Tree construction algorithm to build an FP-Tree. Figure 1 is the constructed FP-Tree. Link is added to speed lookup and easy matching of the pointer to FP-Tree.

![Fig 1: A complete FP-Tree for crime database (DB)](image-url)

#### Step 2: Frequent Itemset Generation

After completing construction of the FP-Tree, the following step of the FP-Growth algorithm is to extract frequent itemsets from the FP-Tree. According to Tohidi and Ibrahim [13] the FP-tree is extracted by dividing the tree (or the compressed database) into sub-databases (conditional pattern base). And from those conditional pattern bases we find out pattern fragments (conditional FP-Tree) associated with each of the databases, and lastly do mining recursively on the tree. Table 3 shows the conditional pattern base, conditional FP-Tree and the mined frequent itemsets from the FP-Tree in Figure 1.
Table 3. Mined Patterns (with ξ=3)

<table>
<thead>
<tr>
<th>Items</th>
<th>Conditional Pattern Base</th>
<th>Conditional FP-Tree</th>
<th>Frequent Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7</td>
<td>{C3, C4:1}, {C3, C5, C6:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C6</td>
<td>[C1, C5:1], {C5:1}, {C2, C5:1}</td>
<td>{<a href="">C5:3</a>}</td>
<td>{C5, C6:3}</td>
</tr>
<tr>
<td>C5</td>
<td>[C1, C2:1], {C1:1}, {C2:1}</td>
<td>{<a href="">C1:2</a>, <a href="">C2:2</a>}</td>
<td>{C1, C5:2, C2, C5:2, C1,C2,C5:2}</td>
</tr>
<tr>
<td>C4</td>
<td>[C1, C2:1], {C3:6}</td>
<td>{<a href="">C3:6</a>}</td>
<td>{C3,C4:6}</td>
</tr>
<tr>
<td>C2</td>
<td>[C1:8]</td>
<td>{<a href="">C1:8</a>}</td>
<td>{C1,C2:8}</td>
</tr>
<tr>
<td>C3</td>
<td>[C1:3]</td>
<td>{<a href="">C1:3</a>}</td>
<td>{C1,C2:7}</td>
</tr>
</tbody>
</table>

2.3 Studies to improve FP-Growth based on single minimum support

FP-Growth algorithm has been blamed to produce large number of conditional pattern base and consequently conditional FP-tree recursively in mining the frequent patterns [14]. Our illustration results in Table 2, which is based on a 20 transactions dataset, can be a good example on this. Studies have thus shown that the algorithm becomes less effective when the dataset size increases. Consequently, several methods have been suggested to improve efficiency of the algorithm. Some of such methods are: implementation of parallel FP-Growth ([15], [14] and [16]), mining only top-k frequent itemsets (Lee and Clifton [17] and Wang et al. [3]), and the use of distributed computing for frequent patterns mining (Deng and Low [18] and Itkar and Kulkarni [19]). Although all of these, and other similar approaches, try to tackle the challenge of the algorithm especially with the increasing datasets, they ignore the mining for infrequent items. In fact, these techniques employ a single user specified minsup. The user specified minimum support threshold assumes that items in the dataset are of identical nature and occurrences. This is however a rare situation in real life applications especially in the crimes datasets where some crime items appear so regularly in the dataset while others appear rarely. It is on this same line that Isafiade et al. [2] used a quartile floor-ceiling functions of the descriptive statistics to propose a pruning step of the FP-Growth. This approach automatically identifies the minsup threshold for the fine-tuning of the algorithm’s pruning step for identifying frequent crime pattern trends. Unfortunately this method works only for small datasets.

2.4 Approaches that use multiple minimum supports

To improve extraction of frequent itemsets studies have proposed the use of multiple minimum supports approach (see for example [26] and [28]). Liu et al. [20] used this approach to mine rare itemsets through an Apriori-like algorithm called Multiple Support Apriori (MSPApriori). According to the author, the approach assigns each item with a minimum support value known as “Minimum Item Support” (MIS). Frequent itemsets are produced under the condition that they satisfy the lowest MIS value amongst the corresponding items.

In this multiple minimum support approach, association rules definition remains the same as presented in section 2.1 above, but the rule’s minsup is defined in terms of MIS of items occurred in the rule. In other words, each item in the database can have MIS value that is calculated using a formula or stated by the user. The provision of different MIS values for different items helps the user to efficiently define distinctive support needs for distinctive rules. For instance, if a dataset consists of four crime items, e.g. murder, robbery, killing_of_albino, and rape, then MIS values could vary as follows: MIS(murder) = 3%, MIS(robbery) = 5%, MIS(killing_of_albino) = 0.1%, MIS(rape) = 0.5%. In addition, the minimum support for any itemset X = (i1, i2, ..., ik), 1 ≤ k ≤ n, is given as

$$\text{minsup}(X) = \min (\text{MIS}(i_1), \text{MIS}(i_2), ..., \text{MIS}(i_k))$$

According to Liu et al. [20] MIS for every 1-itemset (in the MSApriori), expressed as MIS(i), is calculated using the following percentage-based formula

$$\text{MIS}(i) = \frac{(M(i) \times \beta)}{\text{LS}}$$

Where, LS is the least support. This is stated by the user to express the lowest allowed minimum item support, f(i) is the frequency of occurrence of an item in the dataset, and β is the value that governs how MIS values should be associated to their occurrences.

Unfortunately, MSApriori undergoes the same performance drawbacks as the classical Apriori algorithm [11]. FP-Growth-like algorithms that use multiple minsup were then proposed. Specifically, Yu-Han and Yen-Liang [21] proposed CFPGrowth algorithm, and later Kiran et al. [22] proposed the CFPGrowth++ algorithm. According to Kiran [9], the main idea of CFPGrowth++ was the use of the notion of Support Difference (SD) instead of a percentage-based methodology, to specify items’ MIS values as follows.

$$\text{MIS}(i) = \max (\text{Sup}(i) - SD, \text{LS})$$

SD can be either user-specified or calculated from the formula

$$\text{SD} = \lambda (1 - \beta)$$

Where, $\lambda$ is a parameters such as mean, median, and mode of the item, and $\beta$ and LS is the same as for MSApriori.

Although CFPGrowth++ have shown improvements as compared to its predecessors, studies identified that its main weakness is on stating “good” MIS value for each item. According to Chen et al. [23], for example, the algorithm requires users to identify a minimum support value for each item and continuously tune it to obtain the best value. This is a costly in terms of time and efforts.

3. THE PROPOSED APPROACH

Our approach for multiple minimum supports FP-Growth is based on Shannon entropy (also known as Information Entropy). Entropy is simply the average (expected) amount of the information from the event. According to Lesne [24] the Shannon entropy of X is given as

$$E(X) = -\sum_{i=1}^{n} P_i \times \log_2 P_i \quad (1)$$
Where, $E(X)$ (sometimes denoted as $H(X)$) is the entropy of a random variable/item $X$, $n$ is the number of different outcomes, and $P_i$ is the probability of a given item.

We use the Shannon entropy equation (1) to obtain the entropy of each of the crime items in the crime dataset basing on the frequency of occurrence of each of those items. To reflect our context, the Shannon entropy equation is rewritten as shown in equation (2) below. In this equation, $C$ represents crime item.

$$E(C) = -\sum_{i=1}^{n} P(C_i) \cdot \log_2 P(C_i)$$  \hspace{1cm} (2)

This gives the probability of occurrence of a particular crime from a set of similar crimes. In this case, when the number of crime items increase the probability decreases, and thus the entropy. In other words, highly occurring crimes will have higher entropy than the low occurring crimes. To avoid this situation we take reciprocal of the entropy. Reciprocal of the entropy assigns entropy values that increase with the increase of frequency of occurrence of an item. Our entropy equation for crime items thus becomes as shown in equation (3).

$$E(C) = (\sum_{i=1}^{n} P(C_i) \cdot \log_2 P(C_i))^{-1}$$  \hspace{1cm} (3)

The entropy value obtained in equation (3) above gives us the MIS values of crime items in the database. We thus rewrite equation (3) in terms of MIS. We actually replace $E(C)$ with $MIS(C)$ to obtain equation (4).

$$MIS(C_i) = (\sum_{i=1}^{n} P(C_i) \cdot \log_2 P(C_i))^{-1}$$  \hspace{1cm} (4)

Where $MIS(C_i)$ is the minimum item support of crime item $i$ when $MIS(C_i)$ is greater than or equals to $LS$, otherwise $MIS(C_i) = LS$. As Liu et al. [20] defines, $LS$ is the user-specified Least Support. The final MIS of an item will not entirely depend on the value obtained from our aggregate function in equation (4). Depending on the nature of dataset, calculated MIS value can even be one or less than one. We use the concept of LS, where user will set the least support value, to avoid the possibility of getting unreasonable MIS.

Algorithm for obtaining MIS by using this approach is shown in Table 4, and in Figure 2 we show the implementation of the proposed algorithm in Java.

**Table 4. The proposed algorithm for specifying MIS values**

| Algorithm 2: Specifying MIS values using Shannon Entropy |
|-----------------|-----------------|-----------------|
| **Input:**      | **Transaction database (DB), Least Support (LS).** |
| **Output:**     | **Complete set of MIS values** |
| **Process:**    | 1. Scan DB; Count the total number of available distinct crimes in each of the crime categories found in DB. Call the counts $N(C_i)$. |
|                 | 2. For every crime type ($C_i$) compute the probability of ($C_i$) (i.e. $P(C_i)$) as $1/N(C_i)$ |
|                 | 3. Compute the entropy of crime type ($C_i$) as $E(C_i) = -(P(C_i) \cdot \ln(P(C_i)))$ |
| **Output:**     | 4. Compute reciprocal of the entropy obtained in 3 above as $E(C_i)^{-1}$ |
| **Output:**     | 5. If $E(C_i)^{-1} \geq SL$ then $MIS(C_i) = E(C_i)^{-1}$ otherwise $MIS(C_i) = LS$ |

![Java codes for obtaining MIS values using our proposed method](image)

The calculated MIS values are then used to obtain the conditional pattern base and conditional FP-tree from the FP-tree. FP-tree construction uses the same procedures as shown in Algorithm 1 above, but in this case $\xi$ is LS. For example, suppose $LS = 2$, basing on the transaction database in Table1 and the constructed FP-tree in Figure 1, the obtained MIS...
values, conditional pattern base, conditional FP-Tree and the mined frequent itemsets will be as shown in Table 5.

<table>
<thead>
<tr>
<th>Items</th>
<th>MIS</th>
<th>Conditional Pattern Base</th>
<th>Conditional FP-Tree</th>
<th>Frequent Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7</td>
<td>3</td>
<td>{C3, C4:1}, {C3, C5, C6:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C6</td>
<td>3</td>
<td>{C1, C5:1}, {C5:1}, {C2, C5:1}</td>
<td>(<a href="">C5:3</a>)</td>
<td>{C5, C6:3}</td>
</tr>
<tr>
<td>C5</td>
<td>3</td>
<td>{C1, C2:1}, {C1:1}, {C2:1}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>C4</td>
<td>4</td>
<td>{C1, C2:1}, {C3:6}</td>
<td>(<a href="">C3:6</a>)</td>
<td>{C3,C4:6}</td>
</tr>
<tr>
<td>C2</td>
<td>4</td>
<td>{C1:8}</td>
<td>(<a href="">C1:8</a>)</td>
<td>{C1,C2:8}</td>
</tr>
<tr>
<td>C3</td>
<td>4</td>
<td>{C1:3}</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

### 4. PROTOTYPE IMPLEMENTATION

In order to examine effectiveness of our proposed solution we developed a working prototype basing on the suggested approach. The developed prototype allows reporting of crime as well as extraction of patterns from crime data. We call this prototype Crime Reporting and Pattern Extraction System (CRaPES). CRaPES is a desktop application developed by using Java. But since this prototype allows reporting and thus storing of crimes, it comprises of a crime database that was developed by using SQLite.

![Fig 3: CRaPES Dashboard](image)

In running the prototype, we used a MacBook Air (MacOS Sierra version 10.12.4), Intel Core i5 1.6GHz processor machine with 8GB of memory. The prototype allows extraction of crime patterns from crime data stored in CRaPES database as well as from external sources. Since CRaPES database did not have data enough for experimentations, we used external sources of data to evaluate this prototype. Specifically, we obtained our data online from the link https://catalog.data.gov/dataset?tags=crime. CRaPES allows data file to be imported and then MIS values of each of the crime item in the dataset to be calculated based on the method proposed in this paper. Figure 4 is the CRaPES interface for MIS calculation. After calculating MIS values of the crime items, the file containing those values (i.e. MIS file) is exported.
After obtaining MIS file the next step is the actual pattern extraction. In this case, as shown in Figure 5, both data file (crime dataset) and its MIS file are imported to the system and the algorithm is run to obtain the patterns.

We furthermore evaluated how our proposed solution behaves on the varying sizes of crime data in comparison with the existing approaches. For that to be achieved, we created four clusters of input data: the first cluster was 5KB with 847 records (we represent this dataset as DATASET-I), the second was 10KB with 1390 records (DATASET-II), third was 15KB with 2162 records (DATASET-III), and the fourth was 20KB with 2910 records (DATASET-IV). Our evaluation criteria were execution time and memory usage. Thus, we compared execution time and memory consumption on our proposed solution over FP-Growth with varying minimum support thresholds.

Table 6 shows memory consumption in the classical FP-Growth with minimum supports of 10, 20 and 30, and memory consumption with our proposed solution. It was observed that varying user-defined minimum supports did not affect memory consumption of the FP-Growth algorithm, but when the size of the dataset increased our proposed solution was more effective in terms of memory consumption.
Concerning execution time, as shown in Figure 6 we observed an increase of time to complete algorithm’s execution as the size of dataset increased. Our proposed solution, however, recorded a lower execution time as compared to FP-Growth algorithm with minimum support values of 10, 20 and 30.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Memory Use (in MB)</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MinSup=10</td>
<td>MinSup=20</td>
</tr>
<tr>
<td>DATASET-I</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>DATASET-II</td>
<td>2.24</td>
<td>2.24</td>
</tr>
<tr>
<td>DATASET-III</td>
<td>2.88</td>
<td>2.88</td>
</tr>
<tr>
<td>DATASET-IV</td>
<td>3.52</td>
<td>3.52</td>
</tr>
</tbody>
</table>

Fig 5: Execution Time

5. CONCLUSION
In this research paper, we proposed an approach for finding Multiple Itemset Support (MIS) values as an approach to improve rare itemset mining with the FP-Growth algorithm. Specifically, we have proposed an algorithm for obtaining MIS values based on Shannon entropy equation. This approach scans the entire dataset and assigns MIS values on each crime item in the dataset basing on its frequency of occurrence. In this way our approach tackles the rare item problem of the FP-Growth. We tested our solution on varying crime datasets and compare execution time and memory consumption of our solution with classical FP-Growth algorithm. That was achieved through a developed prototype for crime reporting and pattern extraction. Experimental results show that our proposed solution is reasonable and more effective as it outperforms classical FP-Growth in terms of execution time and memory use.

6. ACKNOWLEDGMENTS
We are grateful to the Tanzania Communications Regulatory Authority (TCRA) for financial support that enabled us to accomplish this research work.

7. REFERENCES


