Predictive Trend Mining for Social Network Analysis

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy
by
Puteri N. E. Nohuddin

May 2012

Dedication

To my beloved sons,
Megat Daniel Arif and Megat Ilhan Arif

Abstract

This thesis describes research work within the theme of trend mining as applied to social network data. Trend mining is a type of temporal data mining that provides observation into how information changes over time. In the context of the work described in this thesis the focus is on how information contained in social networks changes with time. The work described proposes a number of data mining based techniques directed at mechanisms to not only detect change, but also support the analysis of change, with respect to social network data. To this end a trend mining framework is proposed to act as a vehicle for evaluating the ideas presented in this thesis. The framework is called the Predictive Trend Mining Framework (PTMF). It is designed to support "end-to-end" social network trend mining and analysis. The work described in this thesis is divided into two elements: Frequent Pattern Trend Analysis (FPTA) and Prediction Modeling (PM). For evaluation purposes three social network datasets have been considered: Great Britain Cattle Movement, Deeside Insurance and Malaysian Armed Forces Logistic Cargo. The evaluation indicates that a sound mechanism for identifying and analysing trends, and for using this trend knowledge for prediction purposes, has been established.

Contents

List of Figures

List of Tables

Acknowledgement

First and foremost, my greatest debt of gratitude must go to my first supervisor, Dr. Frans Coenen. He patiently provided the supervision, encouragement, constructive criticism, research ideas and advice to proceed through the PhD program and complete my thesis. I also would like to express my appreciation to my second supervisor, Dr. Rob Christley, for his support, suggestions and valuable comments.

I am also thankful to Christian Setzkorn. Christian has been helpful in providing advice many times during my research work. I also would like to extend great appreciation to Dr. Wataru Sunayama for his valuable collaboration.

I also extend my gratitude to the Ministry of Higher Education and Universiti Pertahanan Nasional Malaysia for the financial support that has given me the opportunity to undergo the PhD program.

The best and worst moments of my doctoral journey have been shared with many people. It has been a great privilege to spend several years in the Department of Computer Science at University of Liverpool, and its staff members will always remain dear to me. My colleagues and friends, Ayu, Zurina, Zuraini, Stephanie, Santhana, Hanafi, Matt and the rest, thank you for the support and suggestions.

I wish to thank my parents and family especially my beloved sons, Megat Daniel Arif and Megat Ilhan Arif. Their love provided my inspiration and was my driving force to complete my PhD. My sister, Puteri Nor Alaina, thanks to her for all her love, advice and support. Finally, I would like to dedicate this work to Mohd Azrai Najib.

Chapter 1

Introduction

Data Mining (DM) is a generic term used to describe processes used to achieve the automated analysis (by computer) of data with the aim of discovering hidden knowledge [57]. DM is an element in the Knowledge Discovery in Data (KDD) process. KDD encompasses a set of techniques that include, for example, data warehousing, data pre-processing and post-processing; as well as DM. DM has many applications such as:

1. Bank and financial industry data analysis, where it is used to minimize fraud and identify high risk or bad customers [4, 19], and to attempt to forecast stock market movement [119].

2. Medical research, where it is used to monitor (for example) the growth of cancer cell patterns in patients [34].

3. Retail industry support, where it is used to develop marketing and stock replenishment strategies based on customer behaviour and purchasing patterns [14].

4. Telecommunication and computer network analysis, where it is used to identify the loyalty of (say) mobile subscribers in terms of churn rate [46, 100], and to detect network intrusions or irregular behaviour with respect to network users [16, 36].

DM encompasses a variety of techniques such as classification, clustering and pattern discovery. The work described in this thesis is predominantly directed at the latter. In pattern discovery the patterns of interest may take many forms, such as frequently occurring word groups that may exist across a document collection or frequently occurring sub-graphs in graph data. More commonly the frequent patterns of interest are simply frequently occurring sub-sets of attribute values that occur together in tabular datasets. The extraction of frequent patterns from data is typically computationally expensive because, given any reasonably sized dataset, there tends to be a large number of potential frequent patterns. Given n binary valued attributes there are potentially $n^2 - 1$ frequent patterns (minus one to exclude the null pattern).

The amount of data available for the application of data mining has increased rapidly over recent years. Reasons for this include the availability of inexpensive storage and increases in the capabilities of computer hardware. In parallel to the increase in the amount of data collected there has been a corresponding increase in the desire to apply data mining techniques to this data. There is also an increasing interest in studying the spatiality and temporality of the data as this may provide further interesting and useful insights. One element of the latter is trend mining, where we wish to identify how patterns change with time (or do not change).

In the work described in this thesis a trend is conceptualized as a time series comprising a sequence of "occurrence" values plotted against time. More specifically the author is interested in identifying temporal trends in networks such as social and distribution networks. Social networks represent the interaction among individuals in some social setting; the nodes in the network typically represent the individuals and the links the interactions. In distribution networks the nodes describe locations (which might be individuals) and the links the "traffic" between locations. Networks, although typically conceived of in terms of graphs, can also be represented in a tabular format such that each record represents a time stamped "interaction" between two nodes. As such tabular pattern mining techniques can be used to identify patterns in a tabulated "snap shot" of a network. If we then take a sequence of snap shots we can mine trends in the data by identifying changes in the patterns over time. Given that, as noted above, frequent pattern mining typically results in a large number of patterns a significant issue in such trend analysis is the large number of frequent patterns and trends that may be discovered. To address this issue this thesis describes an overall frequent pattern trend mining and analysis mechanism. The proposed framework is designed to identify frequent pattern trends and also provide mechanisms to group and analyse large number of frequent pattern trends. The proposed trend analysis is directed at detecting changes in a sequence of identified frequent pattern trends. This thesis also considers additional analytical techniques, including visualisation and prediction techniques. The visualisation technique provides assistance for users to interpret trend analysis results. Finally the prediction technique uses knowledge of trends to support the investigation of the movement of patterns within social networks.

The rest of this introductory chapter is organized as follows. In Section 1.1 the motivation for the research is discussed. Section 1.2 presents the research question and associated issues. In Section 1.3 the programme of work is outlined. Then Section 1.4 discusses the criteria used to evaluate the research outcomes (the "criteria for success") followed in Section 1.5 with detail of the "contribution" of the research. Section 1.6, then presents an outline of the structure of the remainder of the thesis, followed in Section 1.7 with details of published work produced as a result of the described research. Finally, in Section 1.8 the chapter is concluded with a brief summary.

1.1 Research Motivation

Trends provide useful information for decision and policy makers. For example knowledge of seasonal trends in customer buying patterns, trends describing change in the behaviour of social networking site users, and trends on how some disease or condition might spread in a given geographical area, are all potentially useful to decision makers. The discovery of interesting trends helps us to detect dynamic changes in data that can lead to actions being taken and/or policy or regulation amendments.

The motivation for the research described in this thesis can thus be broadly identified as the desire to realise the advantages that frequent pattern trend mining can offer to support decision makers. More specifically, in this thesis a number of specific applications are considered in detail: (i) The cattle movement tracking system in operation in Great Britain (GB), (ii) an online insurance quote system operated by Deeside Insurance Ltd and (iii) the logistics network operated by the Malaysian Armed Forces. It is suggested that analysis of the GB cattle movement network provides trend knowledge useful for policy and decision makers who wish to monitor and address issues such the spread of cattle disease. Similarly the analysis of the Deeside Insurance dataset provides useful knowledge with respect to customer behaviour to support marketing initiatives. Trend interpretation within the Malaysian Armed Forces Logistic Cargo network provides for logistic item stock management and distribution pattern monitoring over time. Further details of these datasets are presented later in this thesis.

1.2 Research Issues and Question

Given the above the key aim of the work described in this thesis is to research and investigate effective mechanisms to: (i) discover temporal frequent patterns and trends in network data, and (ii) facilitate the analysis of these trends and patterns to predict behaviour across networks. Realisation of this aim requires the solution of a number of research issues:

1. **Frequent Patterns and Trends**: How can we represent frequent patterns and trends so as to facilitate the desired trend mining? How do we transform the raw datasets to support the mining process? Given a large quantity of temporal data, how do we handle the granularity of the time stamps? How do we represent and highlight the trends as a time series result?

2. **Change Detection**: How can we detect changes in the identified trends? How do we define the types of changes that we are interested in? How do we measure the degree of these changes?

3. **Interesting Trends**: Some identified patterns and trends may not be useful to users and stakeholders. Thus how do we handle a large number of generated

patterns and trends? How do we measure the interestingness of these patterns and trends? Can we apply constraints to the data to anticipate interesting, desirable and useful patterns and trends?

4. **Interpretation of Patterns and Trends**: How do we interpret types of frequent pattern trends to the users? How do we annotate the changes occurred in frequent pattern trends in the mining and analysis process?

5. **Prediction**: How can we predict the "percolation" of information within a network? Can we use the discovered patterns and trends to predict the probability of any activity or event in the network? If prediction is possible, what methods are best to manipulate the patterns and trends?

6. **Visualization**: How can we visualize the findings to enhance user understanding? What are the suitable interfaces/features for projecting the results? What methods are best suited to illustrating the temporality of patterns, and trend changes and predictions? If spatial patterns are involved, how can we best relate these patterns to the actual geographical locations?

The overriding research question is thus: *"What are the most appropriate mechanism for identifying, analyzing and displaying trends in network data; and how might those trends usefully be employed for prediction purposes?"* The following section provides a description of the broad research methodology adopted to address this research question.

1.3 Research Methodology

To act as a focus for the work a social network extracted from the GB cattle movement database was used. This was selected because: (i) this provided a substantial network, (ii) it featured time stamps and (iii) analysis of the network would provide an exemplar of the kind of application where the results of the research could be usefully employed. As the research progressed two additional datasets were considered: Deeside Insurance and Malaysian Armed Forces Logistic Cargo. The following programme of work was adopted:

Representation: Investigation of: (i) mechanisms whereby network data could be represented as tabular data, and (ii) mechanisms for conducting the necessary preprocessing with respect to the target datasets.

Frequent Pattern Trend Mining: Investigation into mechanisms to identify the desired trends. The intention here was to build a frequent pattern trend mining system that could be analysed and evaluated, and which could then be used as the foundation for work conducted in latter stages within the programme of work.

Trend Clustering: The identification of some mechanism whereby the anticipated large number of identified trends could be grouped so as to facilitate understanding. The intention here was to use some form of SOM to achieve the desired clustering.

Change Detection: The investigation of techniques whereby changes in trends (or the absence of changes) could be identified. The fundamental idea here was to research mechanisms whereby a sequence of self organizing maps could be related and thus trend movements from one cluster to another identified.

Visualisation: Having identified changes in pattern trends it was felt to be desirable to have some mechanism for displaying this to end users. An investigation into a strategy whereby pattern changes could be visualized was therefore deemed desirable.

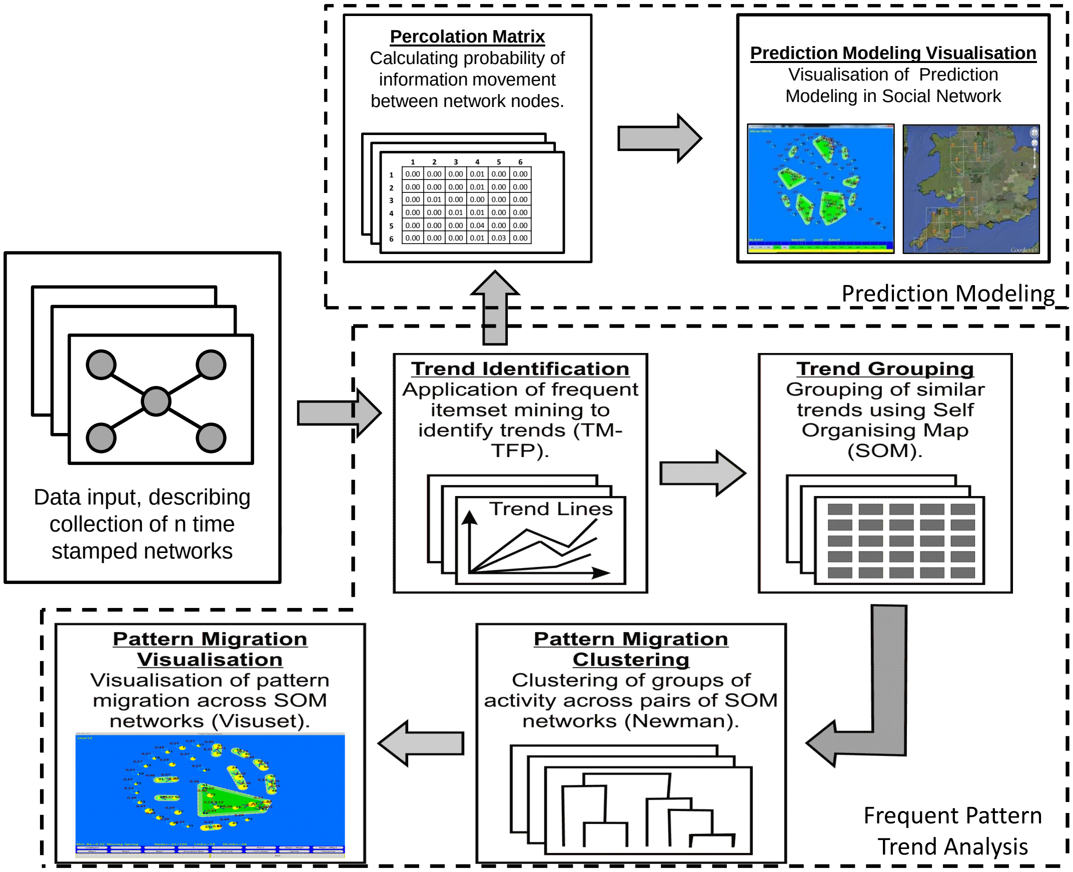
Prediction: Given knowledge of the pattern trends that exist within a network data collection the final phase in the programme of work was concerned with an investigation of how this knowledge might be used to predict the progress of some event across the network.

It was deemed desirable to incorporate the above elements into some forms of integrated framework, a particular artifact resulting from the proposed programme of work is therefore the Predictive Trend Mining Framework (PTMF). Figure 1.1 illustrates the conceptual model of the PTMF which consists of two parts: (i) Frequent Pattern Trend Analysis and (ii) Prediction Modeling. The Frequent Pattern Trend Analysis part has four modules to identify and analyse the frequent patterns and trends that may be contained within a network dataset. The Prediction Modeling part comprises two modules to determine and predict the probability of future activities in a social network.

1.4 Evaluation Criteria

This section discusses the evaluation criteria used to measure the quality of the research undertaken in the context of the above programme of work. The aim was to develop criteria that could be usefully employed to determine the effectiveness of techniques proposed to address the various identified research issues. The following requirements were therefore considered:

1. **Genericity**. Any proposed technique was required to be generic so that it would have general applicability, thus allowing for the analysis of different forms of social network data, from www usage data to business community data. Genericity was demonstrated by applying proposed techniques to a variety of social network data collections.


Percolation Matrix
Calculating probability of information movement between network nodes.
Prediction Modeling Visualisation
Visualisation of Prediction Modeling in Social Network
Prediction Modeling
Data input, describing collection of n time stamped networks
Trend Identification
Application of frequent itemset mining to identify trends (TM-TFP).
Trend Lines
Trend Grouping
Grouping of similar trends using Self Organising Map (SOM).
Pattern Migration Visualisation
Visualisation of pattern migration across SOM networks (Visuset).
Pattern Migration Clustering
Clustering of groups of activity across pairs of SOM networks (Newman).
Frequent Pattern Trend Analysis

	1	2	3	4	5	6
1	0.00	0.00	0.00	0.01	0.00	0.00
2	0.00	0.00	0.00	0.01	0.00	0.00
3	0.00	0.01	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	0.01	0.00	0.00
5	0.00	0.00	0.00	0.04	0.00	0.00
6	0.00	0.00	0.00	0.01	0.03	0.00

Figure 1.1: The Conceptual Model of the Proposed Predictive Trend Mining Framework

2. **Computational time and memory**. Most frequent pattern mining algorithms are computationally expensive. As the size of the dataset increases, the computational and memory resource required increases significantly. Any potential trend mining and analysis technique should therefore be able to process large numbers of records in reasonable time. Run time and memory usage measurements were therefore used as a mechanism for determining the effectiveness of proposed techniques.

3. **Flexibility and Reusability**. Regardless of their specific nature trend mining and analysis mechanisms should be able to adapt to accommodate different types of datasets. For example any proposed algorithm should be able to accommodate further features. It also should be able to accept data attribute selections and data constraints to reflect individual user interests. Users should also be able to conduct the desired trend mining with different levels of granularity; for example weekly, monthly and yearly. Flexibility and reusability was tested using different scenarios (some incorporating constraints) and different granularities.

4. **Scalability**. Any proposed technique, to be considered genuinely useful, should be scalable, i.e. it should be able to operate with large datasets. Thus datasets featuring substantial numbers of records and/or attributes were used to evaluate the proposed techniques.

5. **Accuracy**. Clearly the proposed technique should also discover the correct patterns and trends. This was established, using the cattle movement database, through consultation with domain experts.

As already noted, for evaluation purposes several real world and diverse network datasets were used: (i) GB cattle movement, (ii) Deeside Insurance quotes and (iii) Malaysian Armed Forces logistic cargo distribution.

1.5 Research Contributions

The main contributions of the research work considered in this thesis can be summarized as follows:

1. A mechanism for efficiently generating temporal spatial frequent patterns and trends, that may exist within networks, in terms of episodes or epochs (this will be explained in further detail later in this thesis).

2. A mechanism for clustering groups of trends, using a SOM technique, so as to assist in the further analysis of the identified trends.

3. A trend cluster analysis mechanism to support the detection of changes in trends and frequent pattern migrations.

4. A visualization of pattern movement (traffic) from one trend cluster to another over a period of time, again to facilitate and support trend analysis.

5. A prediction modeling and visualisation technique that can be applied to network data, which illustrates the manner in which information (events) might travel across a (social) network.

1.6 Structure of Thesis

The rest of this thesis is organized as follows:

Chapter 2 presents a literature review of the related research on data mining and KDD, frequent pattern and temporal mining, social network and trend analysis, prediction and lastly visualisation methods.

Chapter 3 presents a brief description of the selected network datasets. As already noted three network datasets were used for the experiments: (i) the GB cattle movement data, (ii) Deeside Insurance quotation data and (iii) Malaysian Armed Forces (MAF) logistic cargo distribution data. The GB cattle movement data has been used as the main dataset for the experiments, the latter two were used to confirm the genericity and flexibility and reusability of the proposed algorithms.

Chapter 4 presents the proposed modules for the Frequent Pattern Trend Analysis which is the first part of the PTMF. This includes mechanisms to generate frequent patterns and trends, cluster similar groups of trends and detect changes in frequent patterns and trends over a period of time. This part of the framework consists of four modules (Figure 1.1), (i) the Trend Identification module to mine frequent patterns and trends using the Trend Mining-Total From Partial algorithm, (ii) the Trend Grouping module to group large numbers of discovered trends to ease the process of trend analysis, (iii) the Pattern Migration Clustering module to analyse the temporal pattern movement from one trend cluster to another and to identify communities of clusters of pattern migrations, and (iv) the Pattern Migration Visualisation module designed to provide a mechanism for illustrating trend changes and pattern migrations to end users.

Chapter 5 presents an evaluation of the proposed modules for the Frequent Pattern Trend Analysis which were introduced in Chapter 4, from pattern and trend identification to the visualisation of pattern migrations. The evaluation was conducted with respect to the criteria identified in Sub-section 1.4 above.

Chapter 6 introduces the Prediction Modeling technique. The technique comprises several elements to predict the "percolation" of information and events in a network given specific frequent patterns. The framework also includes a visualization tool to provide an animation of the percolation. In this case the experiments were performed using the frequent patterns generated using the GB cattle movement data only. A series of experiments were undertaken to demonstrate how the percolation of information and events in a social network can be predicted. A mechanism to support the "drilling down" into trend data is also considered.

Chapter 7 concludes the thesis and presents a summary of the work presented and the main findings in terms of the identified research question and issues. The chapter also includes a discussion on possible directions for a future work.

1.7 Published Work

Some of the work described in this thesis has been the subject of number of refereed publications. These are itemized below.

1. **Journal Papers**

   (a) *Nohuddin, P.N.E., Coenen, F., Christley, R., Setzkorn, C., Patel, Y. and Williams, S. Finding "Interesting" Trends in Social Networks Using Frequent Pattern Mining and Self Organizing Maps. Knowledge Based System Journal 2011.* Journal article comprising an extended, updated and revised version of (f).

   (b) *Nohuddin, P.N.E., Sunayama, W., Coenen, F., Christley, R. and Setzkorn, C. Trend Mining in Social Networks: From Trend Identification to Visualisation. Will be submitted to Expert Systems: the Journal of Knowledge Engineering 2012.* This is an extended version of (g) that includes details of the pattern migrations mechanism.

2. **Conference Papers**

   (c) *Nohuddin, P.N.E., Coenen, F., Christley, R. and Setzkorn, C. Trend Mining in Social Networks: A Study Using A Large Cattle Movement Database. ICDM, Springer-Verlag Berlin, Heidelberg (2010).* Conference paper reporting on some initial work on trend mining that proposed a trend mining mechanism, founded on frequent pattern mining (the TFP-TM algorithm) and clustering, to identify temporal spatial trends in social networks. The work was illustrated using the GB cattle movement database.

   (d) *Nohuddin, P.N.E., Coenen, F., Christley, R. and Setzkorn, C. Detecting Temporal Pattern and Cluster Changes in Social Networks: a study focusing GB Cattle Movement Database. IFIP Advances in Information and Communication Technology, 2010, Volume 340/2010, 163-172 (2010).* This paper built on work described in (a) and included additional work to detect cluster changes in social networks. The GB cattle movement database was again used in the evaluation section. Trend analysis was done using a distance function to highlight temporal cluster change.

   (e) *Nohuddin, P.N.E., Coenen, F., Christley, R., Setzkorn, C., Patel, Y. and William, S. Frequent Pattern Trend Analysis in Social Networks. ADMA'10 Proceedings of the 6th International Conference on Advanced data mining and applications: Part I, Springer-Verlag Berlin, Heidelberg (2010).* The paper described an extension of work described in (d) whereby some constraints were applied to the mining process so as to enhance the trend analysis results. The evaluation section reported on experiments using the Deeside insurance quotation database as well as the GB cattle movement database used previously.

(f) *Nohuddin, P.N.E., Coenen, F., Christley, R., Setzkorn, C., Patel, Y. and Williams, S. Social Network Trend Analysis Using Frequent Pattern Mining and Self Organizing Maps. AI-2010: SGAI International Conference. pp 311-324 (2010).* The paper reported on a technique for identifying, grouping and analyzing trends in social networks using a cluster analysis strategy to identify "interesting" trends. The study focused on two types of network, star networks and complex star networks, exemplified by two real applications: the GB cattle movement database and the Deeside insurance quotation databases.

(g) *Nohuddin, P.N.E., Sunayama, W., Coenen, F., Christley, R. and Setzkorn, C. Trend Mining and Visualisation in Social Networks. AI 2011: SGAI International Conference on Artificial Intelligence.* Conference paper describing an updated trend mining framework to that published previously, the IGCV (Identification, Grouping, Clustering and Visualisation) framework that introduced the proposed visualisation mechanism. Evaluation of its operation was reported using the GB cattle movement network. This paper won the prize for the Best Student Paper award.

(h) *Nohuddin, P.N.E., Coenen, F., Christley, R. and Sunayama, W. Identification and Visualisation of Pattern Migrations in Big Network Data. PRICAI 2012: The Pacific Rim International Conference on Artificial Intelligence (PRICAI).* The conference paper described the Pattern Migration Identification and Visualisation (PMIV) framework which was designed to operate using trend clusters, extracted from large network data using a Self Organising Map technique. The PMIV framework was also used to facilitate the detection of changes in the characteristics of trends over time, and "communities" of trend clusters. Evaluation of its operation was reported using the GB cattle movement network, Deeside Insurance and Malaysian Armed Forces Logistic Cargo networks.

1.8 Summary

In summary, this chapter has provided an overview and background for the research described in the reminder of this thesis, including details concerning the motivation for the work and the research question and issues. It has also provided a brief description of the programme of work, the research evaluation criteria and the contribution of the work. In the following chapter a literature review, intended to provide much more detail regarding the background concerning the research described in this thesis, is presented.

Chapter 2

Literature Review

As noted in Chapter 1, the research described in this thesis seeks to establish an effective mechanism to identify and group trends found in social network data, and also to facilitate the analysis of these trends with respect to network activity. In addition, the research is directed towards the presentation of the analysis using some form of visualisation. This chapter reviews the relevant previous work on which the proposed framework to realise the desired thesis aims is founded.

This chapter is organized into eight sub-topics as follows: (i) Knowledge Discovery in Databases (KDD) and Data Mining, (ii) Association Rules and Frequent Pattern Mining, (iii) Temporal and Spatial Data Mining, (iv) Trend Mining, (v) Clustering Techniques, (vi) Social Network Analysis and Mining, (vii) Prediction Modeling and (viii) Visualization. Each sub-topic is considered in a separate Section. Section 2.1 elaborates on the concept and process of KDD and distinguishes between KDD and Data Mining. Section 2.2 describes the concept of Association Rules and Frequent Pattern Mining (FPM) and reviews several established FPM algorithms. Sections 2.3 and 2.4 focus on mining techniques using temporal spatial data and trend mining; Section 2.4 also considers trend analysis. Section 2.5 describes the concept of Clustering, especially in the context of using Self Organising Maps as a clustering technique, and reviews some related work on cluster analysis. Section 2.6 discusses the concept of social networks, and techniques in social network analysis and mining. Section 2.7 evaluates the concept of prediction modeling in social networks and also considers techniques which have been introduced to predict events or movement of information (and other activities) in a network. Section 2.8 then describes related work on visualization in data mining. Finally Section 2.9 presents a summary of this chapter.

2.1 Knowledge Discovery in Databases and Data Mining

Using current ICT the amount of stored data has accumulated rapidly. Given this amount of data the assumption is that there is valuable hidden knowledge within this data. The suggestion is that discovery of this knowledge may be useful to decision makers and stakeholders. For example, historical customer bank transaction data may be used to rank and assess customers. Banks have employed such procedures for many years as a means of deciding whether or not to approve loans and credit cards. Companies and institutions of all kinds have used similar methods to identify their most valuable customers.

A variety of tools and methods have been proposed to store data to support business applications [38, 55, 109]. Many database and data administrators use Structured Query Language (SQL) and similar tools to maintain and manipulate stored data. However, such database tools are not able to discover non-trivial hidden information or knowledge in data, such as relationships and/or causal data attribute patterns. The identification of such knowledge requires alternative tools; this is the domain of Knowledge Discover in Data (KDD).

In the research described in this thesis the author focuses on KDD techniques, specifically data mining tools, directed at data that has been extracted from social network information. The assumption is that the mining process is done using historical data which has been transferred from some operational database to a data warehouse. In this section the concept of KDD is discussed further in Sub-section 2.1.1 and the KDD process in Sub-section 2.1.2. Data mining, a central element within then KDD process, is then reviewed in Sub-section 2.1.3.

2.1.1 The Concept of Knowledge Discovery in Databases

The terms Knowledge Discovery in Databases (KDD) and Data Mining (DM) have been used interchangeably to describe the process of extracting useful and meaningful information from data. However in this thesis, and in line with many other authors, KDD is defined as the whole process of discovering useful information and knowledge within data, whereas DM is defined as the task within the KDD process where tools and mechanism are applied to identify (mine) the knowledge of interest [38, 41, 42]. The application of KDD is widespread and includes revenue generation, medical and diseases monitoring and the provision of support for “homeland security”. To give three specific examples: [71] describes a health case management system to identify and predict the possible causes whereby a patient may be considered to be a “high risk” patient; [67] describe a KDD system, to support a social program for children, founded on the application of KDD to street crime data in Ethiopia; and [132] apply KDD to a Taiwanese airline passenger database to identify “valued customers”.

KDD incorporates a number of processes, from the preparation of raw data prior to the application of DM to visualization of the final result. The following sub-section describes these processes.

2.1.2 The KDD Process

As noted above, the KDD process encompasses a number of stages. It is generally acknowledged [28, 38, 129] that most KDD applications can be divided into five stages as follows:

1. **Problem understanding**: During this first stage the scope and boundaries of the KDD problem to be addressed are defined. Discussion with end users and decision makers is typically undertaken so as to establish the objectives of the desired KDD. Data is usually collected from various data sources and combined into a single data repository (data warehouse).

2. **Pre-processing**: The collected data typically includes anomalies that need to be corrected or removed to avoid inaccurate results. The pre-processing stage includes the filtering of data records to remove null values, noise reduction and sometimes the anonimisation of sensitive data.

3. **Transformation**: Some of the collected data may have different formats from one another, thus in stage three (where necessary) all data is converted to a standard format.

4. **Data Mining**: In stage four the actual knowledge discovery takes places using some appropriate data mining technique (the data mining stage is considered in further detail in the following sub-section).

5. **Evaluation**: The final stage of the KDD process comprises the analysis of the data mining results (this might include the use of visualisation techniques).

2.1.3 Data Mining

From the above DM can be claimed to be the central activity within the overall KDD process. DM encompasses the use of tools and techniques to discover knowledge in data. DM can result in the identification of patterns, associations and relationships of many forms. Typical DM activities include Association Rule Mining (ARM), Classification, Clustering, Sequencing and Forecasting [38, 50, 130]. Each is considered in some further detail below:

1. **Association Rule Mining (ARM)**: ARM is concerned with the discovery of relationships between data attributes. The frequently cited exemplar application for ARM is supermarket basket analysis (customers who buy eggs are also likely to buy bread).

2. **Classification**: Classification is concerned with decision making, the allocation of a current situation to a category (class). A frequently cited exemplar application is the situation where somebody wishes to decide whether to go out and

play golf or not given the current weather conditions (the classes in this case are "play" and "don't play"). The classifiers used are generated using what are called supervised learning methods in that they require pre-labeled training data.

3. **Clustering**: Clustering is directed at the process of partitioning data records into groups (clusters) that share similar characteristics. In this case the data groups are not known before hand. Clustering techniques are therefore referred to as unsupervised learning methods. It is interesting to note that once a set of clusters have been derived the definition of the clusters can be used for classification purposes.

4. **Sequence Mining**: Sequence mining refers to the process of determining the relationships between data attributes according to some ordering (normally a temporal ordering). The mining process is usually performed using time stamped data.

5. **Forecasting**: Forecasting is akin to classification however, it incorporates a temporal element. Generally, sequence data is used and some mechanism applied to predicts future events.

The work described in this thesis incorporates elements of ARM, Clustering, Sequence Mining and Forecasting; of which ARM (or Frequent Pattern Mining) is the most central. The following section therefore considers ARM in more detail.

2.2 Association Rules and Frequent Pattern Mining

This section describes the ideas behind Association Rules (ARs) and the more general issues associated with Frequent Pattern Mining (FPM). In this thesis FPM is considered to be distinct from ARM, although ARM incorporates FPM. This section is divided into three sub-sections. The first, Section 2.2.1, considers the process of ARM. The second, Section 2.2.2, considers frequent pattern mining and concentrates on the "classic" and most popular FPM algorithms. The work in this thesis is in part an extension of the Total From Partial (TFP) FPM algorithm, and thus the third sub-section considers this algorithm in some detail. For completeness this section is concluded with a review of some more unusual approaches to FPM in Sub-section 2.2.3 and some discussions on applications and concerns of FPM in Sub-section 2.2.4.

2.2.1 Association Rules

ARM is an unsupervised data mining method. As noted above, the main concept of ARM is to discover relations between data attributes that occur frequently together within a dataset [38, 50]. The most well known ARM algorithm is the Apriori algorithm of Agrawal *et al.* [6] which was originally applied to market-basket analysis. The aim was to provide marketing managers with information that correlated certain product purchases so that this information could be used with respect to advertising, store layout, and so on. The ARM problem can be formally defined as follows:

- I is a set of j distinct items (attributes), $I = \{i_1, i_2, i_3, i_4, \ldots, i_j\}$.
- T is a transaction comprising some subset of I ($T \subseteq I$).
- D is a transaction dataset comprising n transactions, $T = \{T_1, T_2, T_3, T_4, \ldots, T_n\}$.

An association rule (AR) is then a relationship $A \Rightarrow B$, where $\{A, B\}$ are subsets of I and $A \cap B = \emptyset$. This should be read as every time a transaction T contains A it will probably also contains B. The set A is referred to as the *antecedent* and B as the *consequent* of the rule. Note that ARs can not always be reversed, $A \Rightarrow B$ does not necessarily imply $B \Rightarrow A$. The support (frequency) and *confidence* (accuracy) of a rule, are used as measures of the potential effectiveness of a rule. The support (*supp*) of an AR is defined as the percentage of records that hold $A \cup B$ with respect to the total number of records in the input data. An AR is said to be *frequent* or *supported* if its support exceeds some user supplied minimum support threshold σ. The confidence ($conf$) of an AR is defined as the ratio of the support for $A \cup B$ to the support for A:

$$conf(A \Rightarrow B) = \frac{supp(A \cup B)}{supp(A)}$$

If the confidence value for an AR is 1 we have a very good AR. An AR is considered to be “interesting” (valid) if its confidence value exceeds some user supplied confidence value τ. ARs are thus typically generated by first identifying frequent itemsets and then using the criteria of support and confidence to discover significant relationships. Although the above discussion concentrates on the support-confidence ARM framework it should be noted that this has its critics and that alternative ARM frameworks have been proposed [99, 102]. However, the support-confidence ARM framework remains the most popular.

Given the above, ARM can be typically thought of as a two stage processes: (i) FPM and (ii) AR generation [44]. The first is the most computationally expensive because, given any reasonably sized dataset, there tends to be a large number of potential frequent patterns. Thus much research work on ARM has been directed at efficient and effective techniques to achieve FPM. The domain of FPM is of particular interest with respect to the work described in this thesis, FPM is therefore discussed in further detail in the following sub-section.

2.2.2 Frequent Pattern Mining

As noted above FPM plays an essential role in ARM. On its own FPM is concerned with finding frequent patterns (frequently co-occurring sub-sets of attributes) in data.

A variety of FPM algorithms have been proposed. With respect to tabular data the majority of these have been integrated with ARM algorithms. Of these the best known, and most frequently cited, is the Apriori algorithm [6]. There a great many variations of Apriori, two variations of note are AprioriTid and AprioriHybrid. Another well known FPM algorithm is FP-growth which is founded on a set enumeration tree structure called the FP-tree. The Apriori, AprioriTid and AprioriHybrid algorithms are therefore discussed further in Sub-section 2.2.2.1, and FP-growth in Sub-section 2.2.2.2. The FPM algorithm adopted with respect to this thesis is the TFP [29, 30], this is therefore discussed further in Sub-section 2.2.2.3.

2.2.2.1 The Apriori Algorithm

The Apriori algorithm operates in an iterative manner by first identifying frequent 1-itemsets and then using these to identify frequent 2-itemsets and so on in a "generate, count-support and prune" loop. An important aspect of the Apriori algorithm (and many other FPM algorithms) is the downward closure property of itemsets which is used to limit the search space. This property states that an itemset cannot be frequent if its subsets are not frequent. Some pseudo code describing the Apriori algorithm is presented in Algorithm 2.1. Given I, a set of itemsets in a transaction dataset D, the algorithm commences by generating the candidate one itemsets I_k (where $k = 1$); then, for each itemset a_i in I_k, the support for each itemset, $a_i.support$, is obtained. For each itemset a_i where $a_i.support < \sigma$ (where σ is some support threshold), the itemset is pruned from I_k. What is left in I_k are the frequent $K = 1$ itemsets. Now the itemsets in I_k are used to generate the I_{k+1} itemsets (thus using the downward closure property of itemsets). The efficiency of Apriori (and similar algorithms) is significantly affected when it is applied to very large datasets (that can not be held in primary storage) as multiple scans through the database will be required. Thus, many modifications to the Apriori algorithm have been proposed, for example AprioriTid and AprioriHybrid [7], to address this issue.

AprioriTid uses a "vertical" representation of the data where each single attribute has a Transaction ID list (a TID list) associated with it. The support for single items is then simply the length of the appropriate TID list. The support for the two itemsets is obtained from a single intersection operations; and so on. Algorithm 2.2 desribes the pseudo code for AprioriTid. The algorithm commences by using the same candidate itemset generation algorithm as Apriori for producing the candidate sets I_k. The algorithm again proceeds in an iterative manner. At each iteration the algorithm scans I_k and obtains the support of the itemsets using intersection operations. Each item in I_k has a TID list associated with. The size of the intersection of the TID lists is then the support for each item in I_k. If the support is $\leq \sigma$, then the itemset will be pruned from I_k. Then the process repeats until there are no more candidate itemsets.

Algorithm 2.1: Apriori FPM algorithm

```
input  : $I, D$ and minimum support threshold $\sigma$
output: $F$
$F$= { };
$k$= 1;
$C_k = I$;
while $C_k \neq \emptyset$ do
    for $\forall c \in C_k$ do
        Count support for $c$ with reference to $D$;
    end
    for $\forall c \in C_k$ do
        if support $c \leq \sigma$ then
            Prune from $C_k$;
        end
    end
    $F = F \cup C_k$;
    $k$++;
    $C_k$ = the set of candidate $k$-itemsets derived from $C_{k-1}$;
end
```

In terms of speed, performance and memory management, reported experiments indicated that AprioriTid outperformed the original Apriori algorithm when generating large k-itemsets [44, 80]. Apriori performs better than AprioriTid in the initial passes but in the later passes AprioriTid had better performance than Apriori. For this reason a combination algorithm was introduced, called AprioriHybrid, in which Apriori was used in the initial passes and AprioriTid in the later passes.

2.2.2.2 The Frequent Pattern-Growth Algorithm

Another established FPM algorithm is the Frequent Pattern (FP)-growth algorithm. While Apriori develops itemsets using a candidate generation method, FP-growth uses a partitioning-based, *divide-and-conquer* method [18, 50]. In common with a number of other FPM algorithms, including TFP discussed in the following subsection, FP-growth uses a *set enumeration tree* structure, the FP-tree, in which to store itemset data. The pseudo code for FP-growth is shown in Algorithm 2.3. The algorithm starts by calculating the support for each single item in I, unsupported items are pruned from the dataset. The remaining items, in each transaction, are then ordered according to their frequency and stored in FP-tree header table. The transactions are then stored in FP-tree. What sets the FP-tree apart from other set enumeration tree structures is that it includes additional links originating from the header table linking tree nodes that feature the same label. The FP-growth algorithm proceed in a depth first manner starting with the least frequent item in the header table. For each entry the support value for the item is produced by following the links connecting all occurrences of the

Algorithm 2.2: AprioriTid FPM algorithm

```
input  : $I$, set of Tid lists, $\sigma$
output: $F$
$F$= { };
$k$= 1;
$C_k = I$;
for $\forall c \in C_k$ do
    Support $c$ the length of corresponding TID list;
end
for $\forall c \in C_k$ do
    if support $c \leq \sigma$ then
        Prune from $C_k$;
    end
end
$F = F \cup C_k$;
$k = 2$;
$C_k$ = the set of 2-itemsets derived from $C_{k-1}$;
while $C_k \neq \emptyset$ do
    for $\forall c \in C_k$ do
        Obtain support for c from the size of the intersection of the TID lists for
        itemsets in c;
    end
    for $\forall c \in C_k$ do
        if support $c \leq \sigma$ then
            Prune from $C_k$;
        end
    end
    $F = F \cup C_k$;
    $k$++;
    $C_k$ = the set of $k$-itemsets derived from $C_{k-1}$;
end
```

current item in the FP-tree. If the item is adequately supported, then for each leaf node a set of *ancestor labels* are produced each of which has a support equivalent to the sum of the leaf node items from which they originate. If the set of ancestor labels is not null, a new FP-tree is generated with the set of ancestor labels as the dataset, and the process repeated. A disadvantage of FP-growth, when finding long frequent patterns, is that many FP-trees may be generated and processed thus introducing additional efficiency overheads. The benefit provided by FP-growth is that the ordering of the 1-itemsets according to their support, and the pruning of unsupported 1-itemsets at the beginning of the mining process, reduces the size of input dataset thus contributing to the efficiency of the approach (although there is no reason why this expedient cannot be applied to other FPM algorithms).

Algorithm 2.3: FP-growth Algorithm- Frequent itemset mining

```
input  : $I, D$ and minimum support threshold $\sigma$
output: $F$
for $\forall c \in I$ do
    Get support for $c$ from $D$;
end
for $\forall c \in I$ do
    if $c \leq \sigma$ then
        $F = F \cup C$;
    end
end
$H$ = Header table of elements in $C$ order in descending support;
$D' = D$ reordered according to ordering of $H$;
forall the $h \in H$ do
    Follow links through FP-tree and obtain support;
    if $h$ support $\geq \sigma$ then
        add to $F$
    end
    $D_{temp}$ = set of items created by following through links;
    Repeat process using $D_{temp}$ ad $D$;
end
```

2.2.2.3 Total From Partial

The Total From Partial (TFP) algorithm is an established FPM algorithm that, like FP-growth, utilizes a set enumeration tree structure for fast lookup purposes [29]. TFP is itself an extension of another algorithm, Apriori-T, which was developed as a more efficient ARM algorithm than straightforward Apriori. Apriori-T uses a reverse set enumeration tree data structure, the Total support tree (T-tree), that facilitates fast "look up". TFP extends Apriori-T by introducing a second tree structure, the Partial support tree (P-tree), in which partial support counts are stored. TFP offers advantages, with respect to generating frequent item sets, in terms of time and storage efficiency; it also provides a good data structure for finding association rules [30]. As noted above the significance of TFP in the context of this thesis is that it is the foundation on which the proposed TM-TFP trend mining algorithm is based (see Chapter 4). TFP is therefore discussed in some detail in this section. The discussion is presented in terms of the generation of the P-tree and the T-tree, the first is discussed in Sub-section 2.2.2.3.1, and the second in Sub-section 2.2.2.3.2.

2.2.2.3.1 Partial support tree (P-tree)

The concept of the P-tree was introduced by Coenen *et al.* in [29, 30]. The P-tree is described as a "preprocessing" tree structure (similar to the FP-tree) into which an input dataset can be translated so that it is stored in a more concise way and at the same time some partial support counting can take place. Figure 2.1 shows an example of how a P-tree is generated. Let $D = \{\{A,B,C\},\{B,C\},\{A,B,E\},\{B,D,E\},\{A,D,E\}\}$. P-tree generation commences with the first record in D. The record $\{A,B,C\}$ is stored, together with its support count of 1, as a single P-tree node. The second record $\{B,C\}$ is stored in a second P-tree node, also with a support count of 1, and linked to the first node so that it becomes a "sibling" of this first node. The next record $\{A,B,E\}$ has a common prefix $\{A,B\}$ with the first P-tree node. This is therefore split into a parent-child pair, with $\{A,B\}$ as the parent and $\{C\}$ as the child (both with a support count of one). Then $\{A,B,E\}$ is added by incrementing the count for $\{A,B\}$ and adding a further P-tree node for $\{E\}$ as a sibling of $\{C\}$ (with a support count of 1). The fourth record $\{B,D,E\}$ shares a leading substring $\{B\}$ with the P-tree node representing $\{B,C\}$. This is therefore split into another parent-child pair $\{B\}$ and $\{C\}$. The fourth record is then included by incrementing $\{B\}$ and adding $\{D,E\}$ as a sibling of $\{C\}$. The fifth record, {A,D,E}, is included in a similar manner by splitting $\{A,B\}$ and including $\{D,E\}$ as a sibling of $\{B\}$.

Usage of the P-tree provides several advantages with respect to the generation of frequent patterns:

1. Faster run times because the counting of pattern support is done partially as the P-tree is constructed.

2. Reduced storage requirements with respect to large datasets where the likelihood of duplicate records and common prefixes are high.

A comparison between the operation of the FP-tree and the P-tree was conducted by Ahmed *et al.* [9]. Despite similarities in their structure Ahmed *et al.* highlighted two distinctions between the two:

1. The FP-tree is a more pointer-rich data structure which leads to a more complicated implementation, whereas the P-tree is simpler to implement.

2. P-tree nodes seek to hold sequences of item sets which are partially closed, while the FP tree nodes hold separate itemsets.

The internal representation of the P-tree presented in Figure 2.1 is given in Figure 2.2. From the figure it should be noted that a P-tree node has four elements: (i) the node code, (ii) the support value, (iii) a reference to a potential sibling node and (iv) a reference to a potential child node.

2.2.2.3.2 Total support tree (T-tree)

The T-tree is used in the second stage of the TFP algorithm where frequent patterns are identified. A T-tree is a reverse set of enumeration tree that is used to store frequent

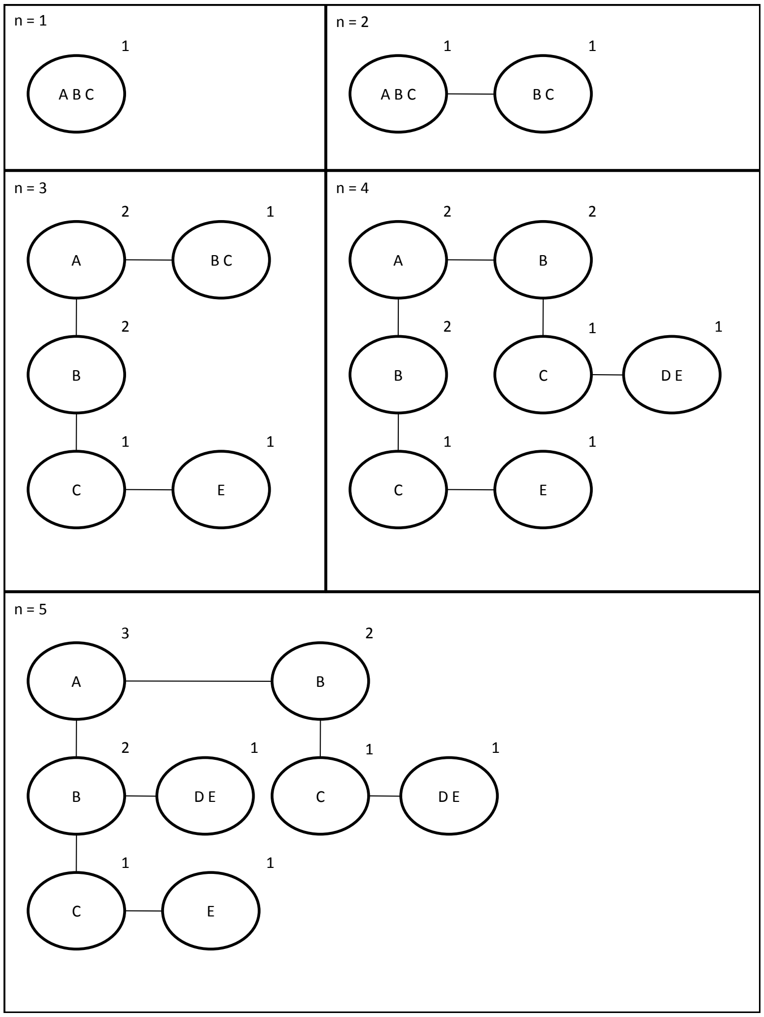


Figure 2.1: P-tree generation

patterns. Each level in the T-tree is actually an array (some authors refer to this structure as a *trie*). Items are stored "in reverse" as this is facilitated by the indexing mechanism permitted by the use of arrays. This indexing also facilitates fast look up [29]. The T-tree is generated in an apriori manner (see above) from the P-tree. In otherwords the T-tree is generated level by level starting with level 1 (one item sets). Figure 2.3 illustrates the T-tree constructed using the P-tree presented in Figures 2.1. The example assumes that the support threshold for frequent patterns is $\sigma = 2$. The T-tree includes nodes for all the items that may exist at a particular level. Initially the support for each node is set to 0. Then, the support counts are updated (Figure 2.3(b)) as a result of a traversal of the P-tree. Then Level 1 pruning is done so that nodes that do not have support above $\sigma = 2$ are "removed". The following level in the T-tree is constructed from the supported nodes in Level 1. Followed by level 2 pruning (Figure 2.3(d)), and so on.

The T-tree data structure provides a number of claimed advantages [9] that lead to

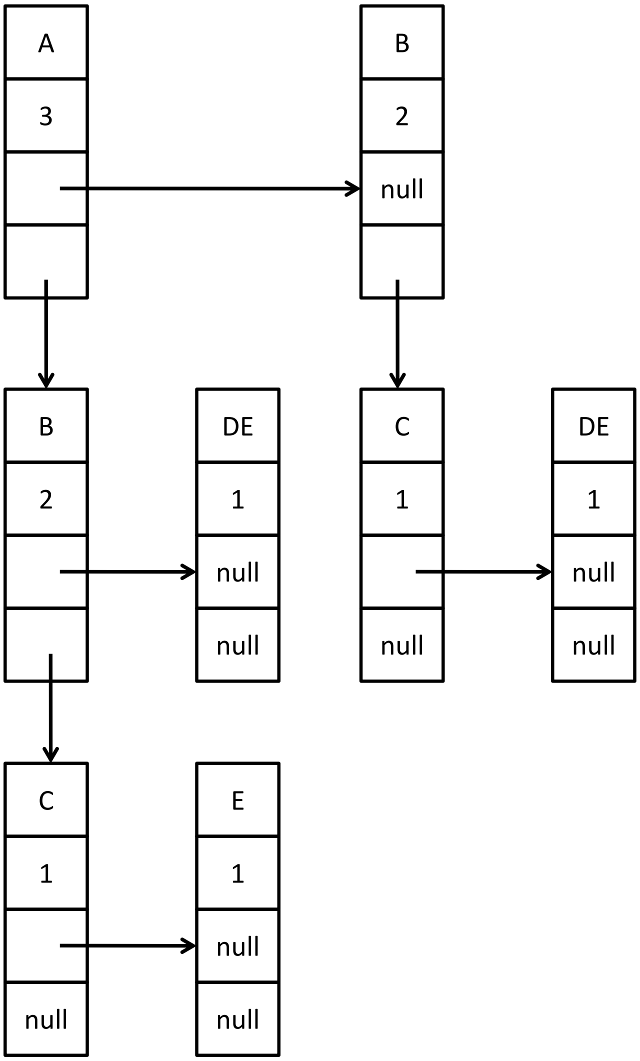


Figure 2.2: Internal representation of P-tree generated in Figure 2.1

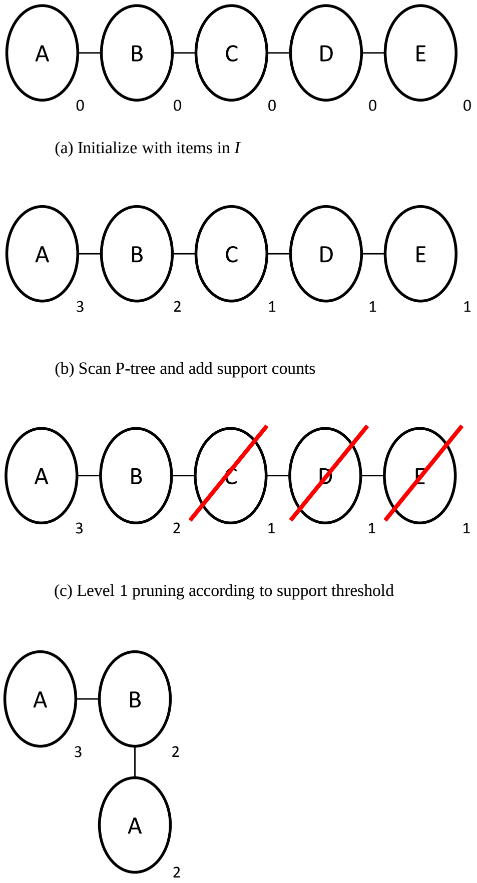
an efficient mining process:

1. The size of the storage requirements for the T-tree is less than that required by other tree structures (such as the FP-tree).
2. The fast lookup facility provided by the indexing mechanism.

Given the benefits of P-tree and T-tree data structures discussed above, the TFP algorithm is used as the foundation for the frequent pattern trend mining algorithm, proposed later in this thesis (Chapter 4).

2.2.3 Alternative FPM algorithms

Many FPM algorithms adopt similar approaches to those described above. From the literature there are also some more unusual approaches and techniques. Examples of



(a) Initialize with items in *I*

(b) Scan P-tree and add support counts

(c) Level 1 pruning according to support threshold

(d) Repeat with further levels until no more candidate sets

Figure 2.3: T-tree generation

note include the use of clustering [33] and linear lists [121]. These two approaches are therefore briefly described in this sub-section so as to complete the discussion of FPM.

In [33] an FPM mining algorithm founded on a clustering method was proposed to identify co-occurrence patterns in data streams. In this technique the data stream was processed using a sliding window of size $k \geq 1$, the algorithm then calculated the support of patterns as it screened the data streams.

Another technique for searching frequent pattern, described in [121], created a simple linear list structure called a Frequent Pattern List (FPL). The proposed FPL algorithm dissected the transaction database into smaller parts without intersection, and then compressed and stored the transactions into the FPL. However, the FPL structure has disadvantages concerned with complexity because of the recursive building of sub-FPLs, and also because it is likely to generate duplicate frequent patterns.

2.2.4 Applications of FPM and Concerns

FPM has been applied in isolation in many subject areas. Exemplars of FPM applications are studies of gene expression data in bioinformatics [10], evaluation of user activity patterns from web logs [56] and assessment of drug reactions from patients [25]. These FPM applications are mainly concerned with the discovery of patterns and their associated frequency in large datasets.

One of the concerns when generating frequent itemsets from a large dataset is that when a low minimum support threshold is set, there is a high probability of having a large number of frequent itemsets many of which may not be interesting in the context of knowledge discovery. However, applying a low minimum support threshold is necessary so as not to miss any unusual interesting frequent patterns. Much research work has been directed at the efficient mining of large frequent itemsets with low support counts. One example is in cancer detection where scientists are looking for abnormal gene patterns which have a low support occurrence in the database [136]. Due to the complexity of this data, Yu et al. discuss the use of emerging patterns and jumping emerging patterns to describe the characteristics of cancer abnormal gene patterns. This is because abnormal gene patterns only appear in cancerous tissues but never occur in normal tissues. Thus, in cancer detection, it is important to monitor the low occurrence patterns in human tissues. With respect to the mining process proposed in this thesis, low support thresholds are used to produce frequent patterns so as to avoid any risk of missing potentially significant patterns and trends.

Besides application with respect to static data, FPM can also be applied to generate frequent patterns from sequence data. There are a number of FPM algorithms that have been introduced to mine patterns from sequence data such as SPACE [139], sequential episodes [85], sequential patterns [8] and GSP [113]. In this thesis, the collection of frequent patterns and support counts is obtained over an ordered sequence of time stamps to describe trends. Trend analysis can then be applied to the detected temporal patterns and trends. Using the advantages offered by the P-tree and T-tree data structures, this research has extended the TFP algorithm to process time series data to generate temporal sequences of frequent patterns. Thus, in the following section several temporal mining and related topics are reviewed and discussed.

2.3 Temporal and Spatial Data Mining

As in the case of data mining in general, advances in data storage mechanisms has also afforded many organizations the opportunity to store significant amounts of temporal and spatially referenced data. Consequently, a range of data mining techniques have been proposed that extend established techniques to address the spatial and temporal elements of data, i.e. spatial and temporal data mining. Spatial data mining is concerned with the idenfication of geographically referenced patterns, while temporal data mining is concerned with the identification of temporally referenced patterns. Temporal-spatial (or spatio-temporal) data mining is a combination of the two. The work proposed in this thesis adapts some of the concepts of temporal mining in that the proposed technique processes a sequence of time-stamped data to study frequent pattern trends. The assumption is also that the social network data of interest will include some spatial or geographic information. Sub-sections 2.3.1, 2.3.2 and 2.3.3 review related work in temporal, spatial and temporal-spatial data mining respectively.

2.3.1 Temporal Data Mining

Temporal or time series data mining, as noted above, is directed at data that includes sequences of events [12]. The main aim of temporal data mining is to discover temporal relationships between items in time stamped data [106]. This then allows for (say) the identification of trends and change points within the data. Many approaches have been explored in the context of temporal data mining. Two common approaches are time series analysis [20, 64] and sequence analysis [139].

In this section the literature concerning work on time series data mining that is focused on the analysis of periodic data and prediction is described. One exemplar, Liu *et al.* [83], proposed an automated time series mining technique to predict and analyze time series in the context of fast food franchise operations data. The study used a model, called the Box-Jenkins seasonal AutoRegressive Integrated Moving-Average (ARIMA) model, to perform analysis and forecasting for inventory management and planning, and potential sales opportunities. The original Box-Jenkins model consisted of an iterative three-stage process of model selection, parameter estimation and model checking [22]. Liu *et al.* improved this original Box-Jenkins model to produce an automatic time series modeling procedure which has been employed to investigate periodic time series. Additionally, the model used an automatic outlier detection and adjustment procedure for both model estimation and forecasting.

In time series analysis, periodic patterns are the patterns that appear in a specific sequence. Han *et al.* [48] proposed a technique for the mining of *partial periodic* patterns in time series databases. They focused on partial periodic patterns which are periodic patterns that appear in a subset of all the time series; the significance is that such patterns will receive less attention when the mining process is applied to the complete dataset. They tested the periodic time series using a drill-down method which repeatedly processed the discovered periodic patterns to see whether these patterns were still periodic at a lower level (done to some smallest subsets of the time series interval). Instead of looking at the periodic patterns, the work in this thesis is directed at frequent patterns trends. Moreover, this idea of “drilling down” into time series data has motivated work on prediction modeling described later in this thesis. Research reported in [140] was directed at detecting frequent patterns in financial time series. This study proposed an automated pattern-spotting technique that utilised various data mining and optimization mechanisms such as neural networks, decision trees, regression, and genetic algorithms.

2.3.2 Spatial Data Mining

Spatial data mining, as the name suggests, is concerned with the application of data mining methods to spatial data. Spatial data is data which has one or more location components so that the individual data objects can be conceptualised as being located in a physical space [38, 75, 111]. In general, spatial data mining has similar objectives to classical data mining. However, spatial data tends not to have the same structure as other data. Thus, a number of techniques have been proposed to explore suitable DM functions to mine spatial data. For example, Ester *et al.* [40] proposed a spatial DM framework that encompasses spatial clustering, spatial characterization, spatial trend detection and spatial classification. These DM tasks process the database according to spatial neighborhood measures that include topological, distance and direction relations between objects. Another study, Ceci and Appice [23], introduced an associative approach to classify spatial data objects. The study compared two approaches: (i) a propositional approach that used spatial association rules to construct an attribute-value representation and perform spatial classification by applying classical classification algorithms, and (ii) a structural approach which used an extension of the Naive Bayes classifier to classify the multi-relational spatial data so as to generate multi-relational association rules.

2.3.3 Temporal-Spatial Data Mining

As already noted above, temporal-spatial data consists of temporal and spatial features that can be subjective depending on the perspective of the data users. In general, data used in temporal-spatial data mining can be categorized as follows [106]:

1. Static: Data that has no explicit temporal-spatial context:

    - Temporality of data is described in terms of a sequence of events.
    - Spatiality of data is presented in general terms describing a location such as high land or wet land.

2. Fully temporal-spatial: Data that includes specific attributes that relate to time and/or location:

    - Temporality of data is described with time-stamped events.
    - Spatiality of data is described with geometrics or geographical locations such as latitude and longitude coordinates.

From the literature it is possible to identify a number of general spatial temporal DM tasks and techniques that may be applied, for example mining frequent temporal-spatial patterns [45] and spatio-temporal classification [138]. Another exemplar, Mennis and Liu [88], used ARM to explore the spatial and temporal relationships within geographic urban growth data. The temporal data, and also the geographic maps, were converted into a tabular form to be processed by conventional Association Rule Mining software. In general, the combination of both the temporal and spatial dimensions adds substantially to the complexity of the data mining task. Thus, traditional data mining techniques may not be applicable, or need to be extended to accommodate temporal-spatial aspects. Table 2.1, taken from [135], provides some details on several tasks and techniques for mining temporal-spatial data. From the table, a DM task, for example to segment or categorise temporal-spatial data, may employ one of a number of clustering and classification techniques such as: cluster analysis, Bayesian classification, decision trees or artificial neural networks.

Temporal-spatial Data Mining task	Descriptions	Techniques	
		Static spatial data	Temporal-spatial data
Segmentation	Clustering Classification	•Cluster Analysis •Bayesian classification •Decision tree •Artificial neural networks	•Temporal extension to clustering •Temporal extensions to classification
Deviation and Outlier Analysis	Finding rules and relationships between attributes over time	•Association rules •Bayesian networks	•Temporal association rules •Temporal extension to Bayesian networks
Trend Discovery	Prediction of lines and curves, Summarising temporal database, Discovering correlations among the events in sequences	•Discovery of common trends •Regression	Sequence mining

Table 2.1: A possible classification of spatial temporal DM tasks and techniques [135]

2.4 Trend Mining

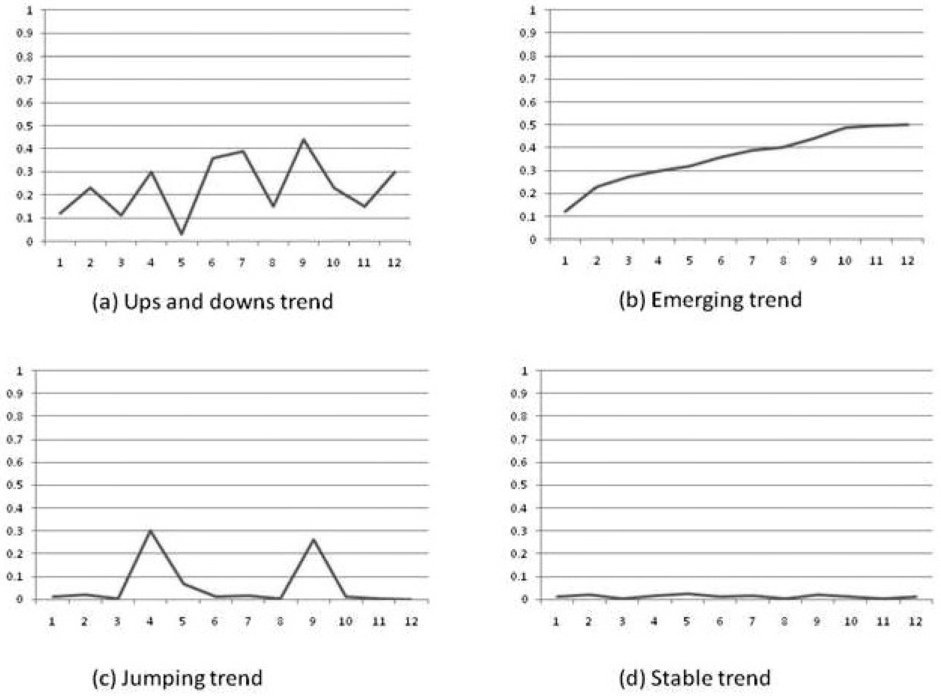
Another perspective of temporal data mining is trend mining. Broadly speaking, trends are indicators of change over time with respect to some activity. The application of DM to time series data allows for the identification of trends and changes in trends. The research described in this thesis is concerned with the identification of temporal-spatial patterns (referred to as "combination patterns") that change over a period of time. Reported work on trend mining has been directed at the forecasting of financial market trends based on numeric financial data, and the usage of text corpi in business news [114]. Other examples of mining trends using temporal and spatial data can be found in biomedical research [84], environmental protection [63] and road traffic management [108]. In this section, Sub-section 2.4.1 discusses previous work on trend mining, followed by Sub-section 2.4.2 which briefly discusses a few related studies concerning trend analysis.

2.4.1 Example of Types of Trend Mining

In the context of this thesis, trends are defined in terms of the frequency counts (support) associated with individual patterns. Given this definition, it is possible to identify some similarities with the work on Jumping and Emerging Pattern (JEP) mining. Jumping patterns are usually defined as patterns whose support changes dramatically from one time stamp to another. Emerging patterns are then a special form of jumping pattern where the support changes from below σ to above σ over two consecutive time stamps. An example of work on JEP mining cam be found in [66], where a moving window approach was adopted to identify JEPs. JEP mining has found application in a number of areas, one example is in medical research where JEPs have been used to monitor the progress of cancer cells [136].

In another study, found in [123], an iterative time-series trend mining mechanism was developed to identify associations in discovered frequent trends using categorical and continuous time-series datasets. The concept of frequent pattern trends defined in terms of sequences of frequency counts has also been adopted in [112] in the context of longitudinal patient datasets. In [112] trends are categorised according to pre-defined prototypes and are grouped using a clustering method. The discovered trends from data episodes in this work were described as *increasing* (emerging), *ups and downs*, *stable* and *jumping* as shown in Figure 2.4.

Figure 2.5 shows an example of a time series of the form considered in this thesis. The time series given in Figure 2.5 is defined by the frequency counts for a pattern X collected over a period t. The granularity of t can be defined as required by the user (for example weeks, months, or years). Clearly from the given trend, useful information can be derived by observing where significant change points occur in the trend line.



(a) Ups and downs trend

(b) Emerging trend

(c) Jumping trend

(d) Stable trend

Figure 2.4: Types of trends

2.4.2 Trend Analysis

Trend analysis refers to the concept of collecting information over a period of time to recognize positive or negative movement or changes in data (where the data typically describes an event or activity). While trend analysis is often used to forecast future behaviors, it may also be used to assess events in historical data. Trend analysis can be valuable as an advanced indicator of some potential problem or issue, for example decreasing trends in sales of a product line.

Trend analysis has been applied in many areas like health [58], climate [128] and human behavior [11]. Examples can also be found in the context of social network analysis. For example, Gloor *et al.* [43] introduced a novel trend analysis algorithm to generate trends from Web resources. The algorithm calculates the values of "temporal betweeness" of online social network node and link structures to observe and predict trends concerning the popularity of concepts and topics such as brands, movies and politicians. Likewise, some research directed at recommender systems [17, 137] and online market research [61] focuses on trends describing online social interactions and trusts so as to improve online marketing and sales strategies.

An important aspect of trend analysis is change point detection (some examples change points were highlighted in Figure 2.5). From the literature, a number of examples mechanism directed at identifying change points can be identified. For example, Taylor introduced a control chart as a change point analysis tool to detect changes in

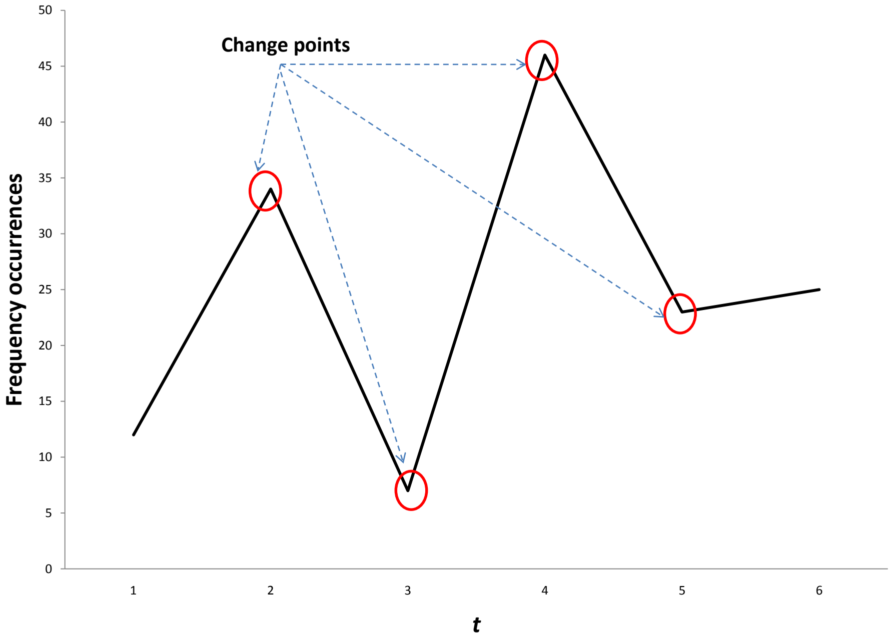


Figure 2.5: Significant Change Points in a Trend.

data and describe the characteristic of these changes [117]. Change point detection has also been considered with respect to a variety of applications. In [39] a system for highlighting change points in historical temperature data was described. In [21] a system was described for detecting change points in Biodiversity measure trends so as to identify species habitat changes.

2.5 Clustering techniques

As mentioned earlier, one of the research issues considered in this thesis is how to analyse large numbers of discovered frequent patterns trends. It is proposed that clustering is a fitting method to categorise large numbers of patterns and trends. Clustering is a commonly adopted method used to group data. This section describes the basic concept of clustering. The approach adopted in this thesis is to use Self Organising Maps (SOMs). SOM technology is therefore considered in Sub-section 2.5.1, followed by a short discussion on cluster analysis in Sub-section 2.5.2.

Clustering is an unsupervised learning method for grouping similar data into "clusters". Clustering has been used in many areas such as biomedical analysis, marketing strategies and environmental monitoring. Clustering algorithms operate in different ways. In some cases the desired number of data clusters is predefined, in other cases the clustering algorithm "works it out for itself". Clustering algorithms typical operate using some measure of similarity, usually a distance measure is used. Clustering technique can be categorised into partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods [50]. Among the most popular clustering algorithms and techniques are K-Means, K-Nearest Neighbour, Hierarchical Clustering and Self Organising Maps:

1. K-Means is a data partitioning technique that divides a dataset into K clusters, where K is predetermined a priori. Each data record belonging to a given dataset is assigned to one of the k-clusters by calculating its distance to the nearest cluster centroid. The algorithm then re-calculates the centroids for the clusters and processes the data again. This procedure continues until the centroids become "fixed".

2. K-Nearest Neighbour is a clustering technique that classifies a dataset by calculating distance between data records. Initially, the first data record, d_1, in a given dataset is used to derive a cluster K_1; then the next data record is considered by determining the distance to the d_1. If the distance is below, τ, a distance threshold, then it will be added to K_1 otherwise a new cluster, K_2, is created. The process continues in an iterative manner until all records are considered.

3. Hierarchical clustering is a method for clustering relatively similar data records or sub-clusters based on measured characteristics, i.e. distance and cohesion, by creating a cluster tree or *dendrogram*. There are two types of hierarchical clustering: (i) Agglomerative (bottom-up), and (ii) Divisive (top-down). The agglomerative hierarchical clustering process begins by considering each record to belong to a single cluster. The two most similar clusters are merged. This merging process continues until a suitable cluster configuration is arrived at (measured in terms of cohesion and separation). The Newman method [91] is an established agglomerative hierarchical clustering algorithm. The divisive hierarchical clustering initially groups the entire dataset into one cluster. This cluster is then split into two clusters. This splitting process continues in an iterative manner, until a suitable cluster configuration is arrived at.

4. A Self Organising Map is a topological clustering tool often used to provide a low-dimensional view of high-dimensional data. It groups the records in a given dataset by assigning records to nodes in the map according to a similarity measure. SOMs are discussed in further detail in Sub-section 2.5.1.

2.5.1 Clustering Trends using Self Organizing Maps

Self Organising Maps (SOMs) were first introduced by Kohonen [73, 74]. Fundamentally, a SOM may be viewed as a neural network based technique designed to reduce the number of data dimensions in some input space by projecting it onto a $n \times m$ "node map" which groups (clusters) similar data items together at nodes. SOMs have been utilized in many research areas. For example SOMs have been used for: clustering gene expression data [125], glaucoma image clustering [134], image retrieval [115] and stock price forecasting [47].

The SOM learning process is unsupervised. Nevertheless, the $n \times m$ number of nodes in the SOM must be prespecified. Currently, there is no scientific method for determining the best values for n and m. However, the $n \times m$ value does define a maximum number of clusters; although on completion some nodes may be empty [31].

The main components of a SOM are the input data (D) and a set of weight vectors (W) to which distance and neighborhood functions are applied to determine which nodes the records in D should be associated with. In Figure 2.6, for example, a SOM map (lattice) comprising 4×4 nodes is presented, each node has n values of weight (w) where n is the dimension of the input data. Each weight has its own unique location in the lattice.

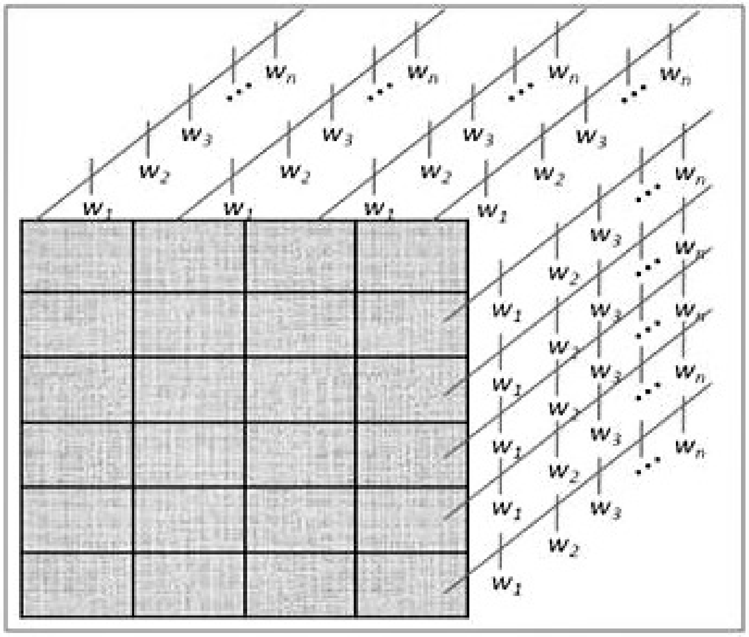


Figure 2.6: A view of 4×4 nodes with n weights

Algorithm 2.4: The basic SOM algorithm

```
input  : $D$, the input dataset
output: SOM
Initialize weights;
for $i \leftarrow 1$ to $|X|$ number of training epochs do
    Get $x$ from $D$;
    Find the "winning" map node for the sample input;
    Adjust the weights of nearby map nodes;
end
```

In the SOM map each record is associated with only one SOM node. Algortihm 2.4 describes the basic SOM algorithm. Firstly, the weight vectors are initialized using a random number generator. Then, the algorithm processes the input data, record by record. For each record, each node "bids" for the record and the record is assigned to the "winning" node. This is done using a distance function. The most common distance function is the Euclidean distance function (2.1):

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.1}$$

where x_i is the i^{th} value of input data and n is the number of dimensions in the input data. If there are two or more weight vectors with the same shortest distance, the winner node is chosen randomly. Subsequently, the algorithm adjusts the neighbouring weights of the winning node to reflect the nature of the most recently assigned record. The most popular mathematical function used in this calculation is based on the Gaussian function (2.2). The magnitude of the adjustment is decreased as the distance from the current node increases.

$$h_{j,i(x)} = e^{\frac{{d^2}_{j,i}}{2\alpha^2}} \tag{2.2}$$

Thus SOM generation is an iterative process; it uses a learning function that decreases with a learning rate ranging between 0 and 1. As the algorithm iterates, it updates (tunes) neighbourhood weights with new values. This adaptive process is based on the function (2.3):

$$w(t+1) = w(t) + \alpha(t)(x(t) - w(t)) \tag{2.3}$$

where w is the selected "excited" weight in the set of topological neighbours, t is the current iteration counter, α is the learning rate in the learning process. Finally, once all records in the training data have been processed, a complete map will be produced representing the records in terms of a set of prototypes (one prototype per node).

With respect to the work described in this thesis, a SOM approach has been adapted to group similar trends, and thus provide a mechanism for analysing social network trend mining results.

2.5.2 Cluster Analysis

This thesis also describes a trend cluster analysis method. Cluster analysis involves observing and recognizing cluster changes in terms of (say) cluster size or cluster membership. There are several reported studies concerning the detection of cluster changes and cluster membership migration. Denny *et al.* [35] proposed a technique to detect temporal cluster changes using SOMs to visualize emerging, splitting, disappearing, enlarging or shrinking clusters; in the context of taxation datasets. Lingras *et al.* [81] proposed the use of Temporal Cluster Migration Matrices (TCMM) for visualizing cluster changes representing e-commerce site usage. As will become apparent later in this thesis, a related idea founded on the concept of migration matrices, will be proposed. Hido *et al.* [53] suggested a technique to identify changes in clusters using a decision tree method.

In the work described in this thesis, as noted above, trends are grouped using a SOM. Each node in the SOM map represents a collection of trends and is referred to in this thesis as a *trend cluster*. The envisioned Predictive Trend Mining Framework considers sequences of trend clusters each represented by a SOM map. By comparing two SOM maps we can see how patterns may move (or not move) between time stamps. This migration of trends can be conceptualised as a (social) network in its own right. To analyse how trend clusters change over time we can apply cluster analysis techniques to these networks. As will become apparent later in this thesis the analysis will be founded on the concept of hierarchical clustering; more specifically the Newman Hierarchical Clustering algorithm mentioned in Section 2.5. Hierarchical clustering is widely used as a cluster analysis tools. Examples of its application in the context of cluster analysis include: identification of the similarity and dissimilarity between cancer cell clusters [93], detecting road accident “black spots” (road traffic cluster analysis) [124] and determining the relationship between various industries based on the movements of financial stock prices [131]. As noted in Section 2.5, Hierarchical clustering can be viewed as a mechanism for identifying communities of clusters according to some similarity value [50].

The following section considers social network analysis and social network mining, the central theme of the work described in this thesis.

2.6 Social Network Analysis and Social Network Mining

This section discusses the concept of social networks and the structure of social network data. A social network describes a social structure of individuals or organisations, who/which are connected directly or indirectly based on a common subject of interest, friendship, business activity or financial exchange. The network depicts the relationship between the social entities, which normally comprise “actors”, who are connected through ties, links or pairs [70, 126]. A social network may depict informal and/or formal connections/communications between actors.

Using social networking sites such as MySpace, Facebook, Twitter, YouTube and Linkedln, individuals are able to “hook up” with one another is an effective manner. Figure 2.7 provides a network traffic report by ComScore Inc. on the number of users of Facebook and Myspace in the United States of America between August 2005 and May 2011 [82]. The figure indicates that social network site usage has increased over the last 5 years. In June 2006, a group study found that 240 million people were using Microsoft Instant Messenger to conduct 30 billion instant messenger conversations. The result was that the average path length of the conversations among the anonymized users explored was 6.6 [79].

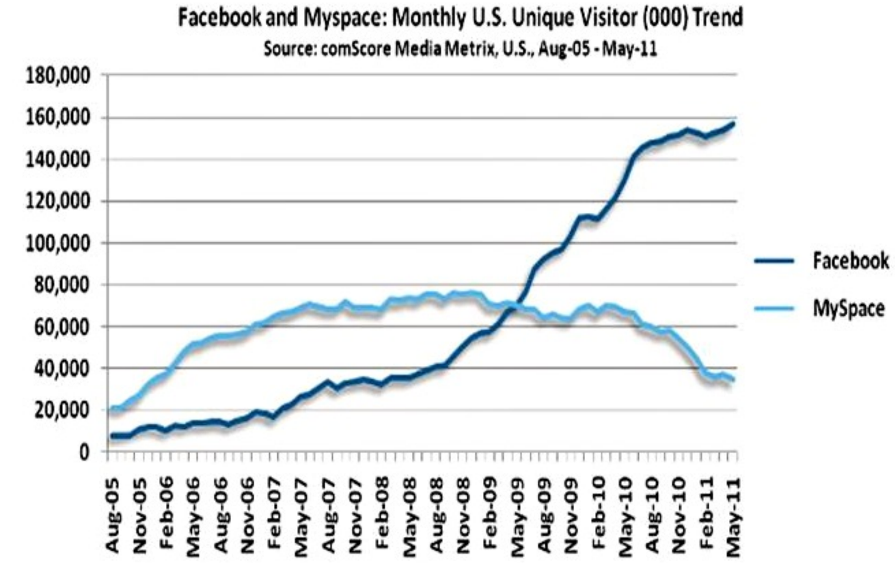


Figure 2.7: Comparison of Facebook and MySpace growth [82]

Social networks are not limited to social networking sites. In the wider context social networks can include business communities, file sharing systems and co-authoring frameworks. The work described in this thesis considers social networks according to the widest interpretation of the term. In the following two Sub-sections further discussion is presented concerning: (i) work related to social network analysis (Sub-section 2.6.1) and (ii) social network mining techniques (Sub-section 2.6.2).

2.6.1 Social Network Analysis

Inspired by the theory of Six Degrees of Separation [69], researchers have started to explore the networks or relationships between people built through friendship, society and economic factors. To analyze the network structure, many social network analysis techniques have been proposed which map and measure the relationships and flows between actors.

Social Nework Analysis (SNA) is the study of social networks with respect to their structure and behaviour [87]. The intention of SNA is to map and measure the relationships and flows between actors in the network. As described above, the principal components in a social network are the nodes and the relational ties. Nodes in the network typically represent people, corporations or groups, while relational ties illustrate relationships between the nodes. SNA has gained prominence due to its practicality in identifying relations within social network data with respect to many application areas such as: marketing, organization management, and spread of disease. SNA utilizes network data which contains at least *structural variables* measured over a set of actors [70, 126]. Structural variables describe quantities that measure the social network structure; for example the "strength" of the relationships between actors. Structural variables can be used to dimension a specific relationship between pairs of actors, for example the degree of friendship or communications between actors. Alternatively, the analyst may consider composition variables which are measurements of actor attributes like gender, age or race. Depending on the application, network data analysts may use a variety of variables from the network datasets.

Social network analysts often use graphs and matrices to visualize the information represented by patterns of nodes and links [51, 126]. The methods of projecting the graphs and matrices are typically adopted from mathematics and are referred to, in the context of social networks, as sociographs or sociograms and sociomatrices or adjacency matrices. A sociograph (sociogram) normally displays network drawings that consist of points (nodes) to represent actors and lines (or edges) to represent ties or relations. In the sociograph the positions of the nodes are unconstrained by coordinates, and the distances between the nodes and angles between the lines of a network drawing are insignificant. Sociomatrices (adjacency matrices) describe relations between actors in tabular forms. If there is an intersection of actor A (in a row) and actor B (in a column), the matrix is marked as "1" which indicates there is an "incidence" between the two actors otherwise "0" is recorded. Knoke and Yang [70] described several other methods for analysing networks, apart from graphs and matrices. These methods include using relationship measures, centrality, visual displays and block models to analyse the actor-link structures.

In another SNA approach presented in [51] the analysis is conducted on an entire population, rather than randomly sampling the network data. This is done through observing the frequency of interaction or intensity of links among the actors, such as observing the volume of link "traffic". Since SNA focuses on relationships among actors, actors cannot be sampled independently; for example, if an actor, X, is selected, then all other actors who are linked to X must also be selected as members of the population.

There are several types of social networks which can be identified. The category of a network is determined by the complexity of the sets of actors and properties which link them together. Social network analysis can be described in terms of the mode, which means the number of distinct categories of social entity in the social network [126]. A one-mode network involves measurements on just a single set of actors, a two mode network involves measurements over two sets of actors or a set of actors and a set of events. A diversity of techniques have been developed to study social networks.

Social network analysis has been proposed as a key technique to examine the interests and issues in sociology subject areas. The majority of social network analysis tools are based on traditional statistical and mathematical models to measure the structure of a given social network. A recent research area is social network mining, where the objective is to find hidden knowledge in social network data.

2.6.2 Social Network Mining

There has been a rapid increase in interest, within the data mining community, regarding the mining of social networks. Because of the demand, a number of Social Network Mining (SNM) techniques have been introduced to identify knowledge concerning the social behaviour of users on online environments [32, 101]. Data mining based techniques are proving to be useful for the analysis of social network data, especially with respect to large datasets.

Originally, social network mining approaches tended to be founded on graph mining techniques; but recently classification, clustering and link mining have also gained popularity. Typical social network mining applications include: (i) the discovery of disease spreading patterns from dynamic human movement [76], (ii) the monitoring of users' topics and roles in email distributions [87], and (iii) the filtering of product ratings from online customer networks for marketing purposes [37].

There are several case studies that describe the modification of traditional data mining methods for application to social network data. There are examples of supervised and unsupervised data mining methods that have been modified to suit richly structured social network data. For example in the case of mining email social networks, researcher have applied clustering algorithms to categorize the mailings, in an open-source software public forum, in order to identify the communication and coordination activity patterns of the participants on exchanging source codes [15]. The clustering algorithm used email IDs and clustered them as distinct email personalities despite the possibility that some email users may have several aliases.

SNM can be applied in a static context, which ignores the temporal aspects of the network; or in a dynamic context, which takes temporal aspects into consideration. In the static context the analysis is directed at either: (i) finding patterns that exist across the network, (ii) clustering (grouping) sub-sets of the networks, or (iii) building classifiers to categorize nodes and links in a "snapshot" of the network. In the dynamic context, a different or extended kind of analysis can be applied to identify relationships between the nodes in the network by evaluating the spatio-temporal co-occurrences of events [78]. A central element of the work described in this thesis is directed at mining social networks in the dynamic context, specifically in terms of the trends and change points that may exist within social networks.

2.7 Prediction Modeling

There are studies on how information moves across a network, for example how neighbouring nodes influence viral marketing [104]. These studies have shown how information percolates from one node to another. As a result, activity spread across a network can be predicted. Another main objective of the research described in this thesis is prediction modeling in social networks. In the following sub-section, the concept is defined and reviewed in comparison to other related studies (Sub-section 2.7.1). Sub-section 2.7.2 then reports on a number of studies for prediction modeling in social networks.

2.7.1 Prediction and Data Mining

In data mining, prediction is a similar process as classification. However, prediction does not necessarily require a categorical class label to be included in the attribute set [50]. Prediction modeling is a significant analytical task concerned with predicting the probability of a future event or trend based on current and historical data [1]. Prediction modeling has been used in several application domains such as the prediction of disease spread, customer relationship management and social media. For example Kiss and Bichler [68] have conducted work directed at phone call and text messaging networks to propose a predictive model to improve customer relationship management. Christley *et al.* [26] studied centrality measures in a simple random social network to determine the probability of infectious animal diseases. Tseng and Lin [122] studied data streams from mobile web systems to generate user behaviour patterns and make predictions based on user location and requested services.

2.7.2 Prediction Modeling in Social Network

Neville and Frost [90] identified two types of social network modeling: descriptive and predictive modeling. Descriptive modeling views social relationships, in terms of network theory, as consisting of nodes and ties (edges, links, or connections), and also categories of social communities that may exist within a network. Whereas, predictive modeling is concerned with methods to analyse the changes in links or edges in a network, and also predict information (feature/attributes) interchange across a network. A number of studies have proposed techniques to predict how social networks behaviour may change over time based on historical data activity. For example the predicting and profiling of the behaviours of online bloggers for application in recommender systems [24]. Taskar *et al.* [116] experimented using relational Markov Network algorithms to predict relational entities and link relationships in networks. Lampos and Cristianini [77] presented a technique to track flu related infection in the UK using the content of Twitter. Their method scanned the content using textual markers and then compared the statistical result with the Health Protection Agency's flu rates. However, the method needed to filter the text to remove instances of media hype or discussion rather than actual flu cases. In another setting, Backstrom *et al.* [13] introduced an algorithm that predicted the relationship between geographical location and friendship among Facebook users. The algorithm confirmed (not unexpectedly) that when the distance between locations increases, the likelihood of friendship decreases. This thesis proposes a prediction modeling technique to indicate the probability of particular information or events traveling across a network; for example how animal disease might spread across a network.

Several studies have proposed prediction modeling for social networks using techniques such as regression analysis, decision trees and Bayesian networks. As noted in the foregoing section, social network analysis was first conducted from a static perspective. Subsequently, researchers have explored the dynamic behaviour of networks. The dynamic evaluation of social networks is conducted by identifying the changes (e.g. trends) that occur across a given network, such as an increase in the number of edges and nodes. Xiang *et al.* [133] developed an unsupervised model to estimate relationship strengths in networks by studying user similarity and interaction. Another interesting prediction model, proposed in Khan [65], introduced a "predictive matrix" for predicting efficient sales based on several characteristics such product colour, customer gender and sales season. The matrix is built after extracting frequent patterns from sales data, and the prediction is made based on the frequency counts of combinations of frequent patterns. The work described in this thesis also uses frequent patterns and frequency counts to predict events; however, the frequency counts are collected in terms of trends. Thus, prediction modeling by considering identified trends so as to predict future activities in a network.

2.8 Visualisation

The term "visualisation", as used in this thesis, refers to techniques to illustrate the findings of data mining activities. Visualisation often acts as a powerful analytical tool to communicate and present data mining results. This section describes some related work on visualisation in DM in Sub-section 2.8.1. Then Sub-section 2.8.2 introduces the Visuset visualisation tool, which was extended to support the work described in this thesis.

2.8.1 Visualisation in DM

The objective of visualisation tools in DM is to support user understanding of the end results. The output of DM processes can be complicated to comprehend. One of the often identified issues in DM that needs to be addressed is the visualization of discovered knowledge [27]. According to a survey conducted by Rexer Analytics [3], a group of data miners noted that they faced challenges in explaining the essence of DM results. Han and Gao [49] also pointed out that effective and efficient visualization tools should be investigated further to determine how they might support the analysis of DM results.

It is generally acknowledged that the visualization of DM results should serve to enhance the users' understanding [118]. The users' need to understand the context of the discovered "hidden" knowledge within the datasets, for example what is the relationship and causal association between attributes? Thus, the visualisation or representation of data mining output should be meaningful. Also, users should be able to interact with the visualisation so as to get further clarification of the results. There are a number of DM software systems, for example WEKA, that include facilities to allow data miners to visualise DM results [130]. MineSet is a DM visualisation tool developed by Silicon Graphics in 1996 [72]. The JUNG programming toolkit was introduced to provide visualisation support for social network mining [89]. JUNG stands for Java Universal Network/Graph, and is a Java library that provides several algorithms that allow social network miners to visualize dynamic graphs by adding or removing nodes and links.

There is also some reported work on data visualisation of temporal data and trends [5] and cluster change [35]. Jung [59] proposed a technique for the visual illustration of recommender system to help users to make more effective decisions. Another study, Rossol *et al.* [107], recommended the usage of a 3D framework for real-time geospatial temporal visualization by evaluating livestock movement data for tracking and simulating the spread of epidemic diseases. The significance of the latter is that live stock tracking is one of the exemplar applications considered in this thesis.

The work described in this thesis, implements three visualization methods for: (i) the visualisation of large numbers of frequent pattern trends in terms of trend clusters, (ii) visualising communities of clusters in the context of trend migration and (iii) visualizing the predicted migration of trends. This thesis utilizes Visuset to illustrate the outcomes from the proposed social network mining.

2.8.2 VISUSET

Visuset is a 2-D visualization software tool that was developed for chance discovery [86]. It represents node communities, using a 2-D drawing area, based on the Spring Model [60]. It highlights which nodes are connected directly and indirectly with other nodes in detected communities which are depicted as "islands". Nishikido *et al.* [94] presented Visuset as an animation interface to illustrate change points in keyword relationship networks. This was considered to be a chance discovery tool because it discovered significant candidates (keywords) that benefited the utilization and selection process. Visuset provides a clear animation of communities of clusters to highlight which clusters connect to which clusters. The significance of Visuset is that the research described in this thesis extended Visuset to support trend cluster analysis and visualisation of significant dynamic cluster changes in sequences of data.

There are alternative visualisation tools that could have been adopted with respect to the work described in this thesis. For example, Kandogan [62] developed a system to display multi-dimensional data in a two dimensional surface as a scatter plot. However, no indication was given of the inter relationships between data points. Visuset groups data into "islands", data within an island is closely linked according to co-relationship values. Visuset thus highlights the nature of the groupings that exist and how the data is correlated. Havre *et al.* [52] described a technique for displaying thematic changes as river flows, so that changes of topics can be observed. However, unlike Visuset, the relationships between topics are not considered. Chen [25] described a system to visualize a network so as to identify emerging trends. However, the network is displayed with respect to a specific time stamp. Therefore changes in trends cannot be easily observed. The extension of Visuset described later in this thesis illustrates trend transitions as an animation so as to demonstrate how trends change over a given period. Robertson *et al.* [105] introduced a system to also show trends by animation. Their method illustrated changes in the data in the form of traces, but changes are considered independently. In the proposed Visuset approach, trends are correlated against one another so that observers can see how groups of trends change with time.

2.9 Summary

This chapter has presented an overview of work related to the general concept of KDD and DM, Association Rules and FPM, temporal spatial DM, clustering and trend cluster analysis, social network mining and visualization in DM. The related work in DM techniques, such as FPM, provided several insights to the proposed module for identifying temporal frequent patterns and trends. TFP was selected as the foundation algorithm and was extended to suit the nature of sequences of social network data. As noted in many FPM experiments, large numbers of patterns are typically discovered which tends to hinder the user's interpretation of DM results. Thus the use of clustering techniques and visualisation tools are proposed. Trend analysis is aimed at investigating temporal changes that occur in collections of frequent pattern trends. In the work described in this thesis, prediction modeling is proposed. The next chapter introduces the modules for the Frequent Pattern Trend Analysis element of the proposed framework.

Chapter 3

Social Network Datasets

This chapter describes the "social network" datasets used for evaluating the algorithms in this thesis. The datasets were extracted from: (i) the GB cattle movement database (ii) an insurance company (Deeside Insurance Ltd) customer database describing requests for insurance quotes and (iii) the Malaysian Armed Forces logistic cargo distribution database. These datasets are exemplars of business community social networks representing the entities that form part of the organisations communities and the traffic/communication between these entities. As noted in Chapter 2, in this thesis, the definition of the term "social network" is extended beyond the "tight" definition used by some authors, namely that social networks represent user of Internet sites such as Facebook and LinkedIn and the communication between those users. This thesis takes a much wider view that social networks may include business communities, file sharing systems, co-authoring frameworks and so on. The selected datasets consist of attributes which are viewed as network nodes for example farms, customers and camps; and movement, communication or traffic between these nodes are treated as the edges of the networks.

This chapter introduces the three datasets used for the evaluation described in later chapters. So that the social network datasets can be used with respect to the systems described in this thesis it was first necessary for them to be preprocessed and appropriately formatted. This chapter thus also explains the discretisation and normalisation processes that were applied to the datasets to produce the required binary valued format.

With respect to the social networks to which the proposed mechanisms may be applied, two specific "type" of social network can be identified. The generic nature of these networks is presented (in a stylized form) by the two "network snap shots" given in Figures 3.1 and 3.2. With reference to Figure 3.1, the network is characterised by a single "star shape" with all nodes communicating with one *super-node*, the author refers to this type of network as a *star network*. Note that, as shown in the figures, not all network nodes will be necessarily communicating (linking) with the super-node at any given time stamp. The generic network snap shot given in Figure 3.2 is a more complex version of that given in Figure 3.1, and in this thesis it is referred as a *complex star network*. The network is characterised by a number of disconnected "star" sub-networks of varying size. Again, not all network nodes (with respect to the snap-shot time stamp) are necessarily communicating (linking) with any of the other nodes. Note also that, some of the "star" sub-networks comprise only two nodes.

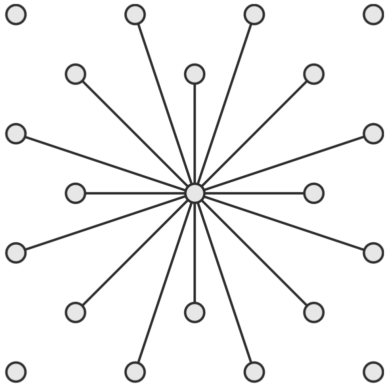


Figure 3.1: (Styalised) Simple Star Network

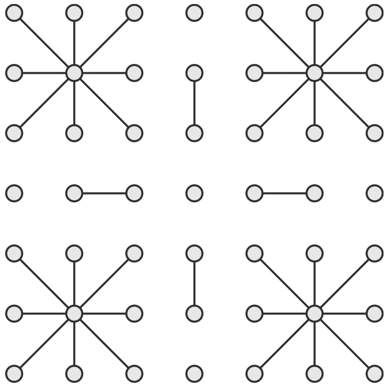


Figure 3.2: (Stylaised) Complex Star Network

The rest of this chapter is organised as follows. Section 3.1 describes the GB cattle movement dataset, Section 3.2 the insurance quotation dataset and Section 3.3 logistic cargo distribution dataset. The discretisation and normalisation process is then presented in Section 3.4, where the data schema for the pre-processed datasets is also explained. Lastly, Section 3.5 briefly summarizes this chapter.

3.1 GB Cattle Movement Database

The GB cattle movement Cattle Tracing System (CTS) database records all the movements of cattle registered within or imported into Great Britain. The database is maintained by the Department for Environment, Food and Rural Affairs (DEFRA). Cattle movements can be "one-of" movements to final destinations, or movements between intermediate locations. Movement types include: (i) cattle imports, (ii) movements between locations, (iii) "movements" in terms of births and (iv) "movements" in terms of deaths. The CTS was introduced in September 1998, and updated in 2001 to support disease control activities. Currently (2012), the CTS database holds some 155 Gb of data.

The CTS database comprises a number of tables, the most significant of which are the animal, location and movement tables. For the analysis reported in the thesis, the data from 2003 to 2006 was extracted to form 4 episodes, each comprising 12 (one month time stamps), presented as a sequence of 48 "complex" networks. The data was stored in a single data warehouse such that each record represented a single cattle movement instance associated with a particular year (episode) and month (time stamp). The number of CTS records represented in each episode was about 400,000. Each record in the warehouse comprised: (i) a time stamp (month and year), (ii) the number of cattle moved, (iii) the breed, (iv) the senders location in terms of easting and northing grid values, (v) the "type" of the sender's location, (vi) the receivers location in terms of easting and northing grid values, (vii) the "type" of the receiver's location, and (viii) the senders' and receivers' Parish Testing Interval (PTI)[1]. If two different breeds of cattle were moved at the same time from the same sender location to the same receiver location, this would generate two records in the warehouse. The maximum number of cattle moved of the same breed between any pair of locations for a single time stamp was approximately 40 animals. The spatial magnitude of movement between farms or animal holding areas can be derived from the location grid values. The easting and northing values of sender and receiver locations were divided into k kilometer sub-ranges to produce k sized grid squares Experiments using $k = 50$ and $k = 100$ were conducted; these are described in Chapter 5. The effect of this ranging was to sub-divide the geographic area covered by the CTS database into a $k \times k$ grid. These grid squares were given unique ID numbers which were also recorded in the dataset.

3.2 Deeside Insurance Database

The Deeside Insurance quotation dataset (provided by Deeside Insurance Ltd, Deeside, UK) was extracted from a sample of records taken from the customer database operated by Deeside Insurance Ltd. Twenty-four months of data were obtained comprising, on average, 400 records per month. The data was processed to produce a sequence of 24 networks, one per month; divided into two episodes comprising 12 months each, 2008 and 2009. Each record consisted of 13 attributes: (i) Aggregator ID[2], (ii) year of insurance contract, (iii) customer gender, (iv) make of car, (v) car engine size, (vi) year of manufactured, (vii) customer postcode, (viii) driver age (ix) conviction code, (x) conviction code number (xi) length of disqualification, (xii) fault and (xiii) penalty (note that the value for some of the attributes may be *null*). The data can be viewed as representing a simple "star" network with Deeside Insurance at the center as a *super node* and all other nodes radiating out from it. The first three digits of the customer postcodes were used to represent geographical locations (the outlying nodes in the star network). The links were labeled with the number of requests for insurance quotes originating from a given geographic location. Each month comprises about 800 records.

[1] PTI is the default frequency for routine TB testing for all cattle herds situated in a parish (a GB geographic unit historically marking out the jurisdiction of a single priest, but frequently used for local government purposes). The PTI will vary from parish to parish accosting to circumstance.

[2] An aggregator is a web application or search facility that allows users to obtain and compare a number of item quotes/prices.

A total of 24 networks were extracted, one for each month covering the period 2008 to 2009.

3.3 Malaysian Armed Forces (MAF) Logistic Cargo Distribution Database

The Logistic Cargo Distribution dataset describes the shipment of logistics items for the Malaysian Army, Air Force and Navy. The logistic items include vehicle, medicines, military uniforms, ammunition and repair parts. The dataset was extracted from the records for 2008 to 2009 to form 2 episodes with 12 time stamps each, thus a total of 24 networks. As in the case of CTS database the extracted data network also described a complex star network as it comprised many simple star networks. Items are sent from a number of division logistic headquarters to brigades and then to specific battalions in West and East Malaysia. The location of headquarters, brigades and battalions are the spatial attributes of the dataset. These offices are viewed as being sender and receiver nodes (in a similar manner as described for the CTS dataset) and the shipments as links connecting nodes in the network. Each month consists of some 100 records. Each extracted record has 7 attributes: (i) time stamp (month), (ii) logistic item, (iii) sender, (iv) sender city, (v) receiver, (vi) receiver city, and (vii) shipment cost.

3.4 Discretisation and Normalisation

Discretisation and normalisation processes were used to covert input data, presented in some non-binary format, into the binary valued format. This was necessary because the data mining techniques to be used, for FPM, will only operate with binary valued data (0-1 data). Discretisation converts the original dataset attributes with continuous data values into $\{1 \ldots N\}$ sub-ranges such that each sub-range is identified by a unique integer label. Normalisation converts data attributes with nominal values into unique integer labels/columns. For the experiments in this research, the attributes with continuous data types were divided into 10 sub-ranges and the attributes with integer data types were divided into 5 sub-ranges. Thus, the data format conversion maintains the nature of the data while at the same time permitting the application of FPM algorithms.

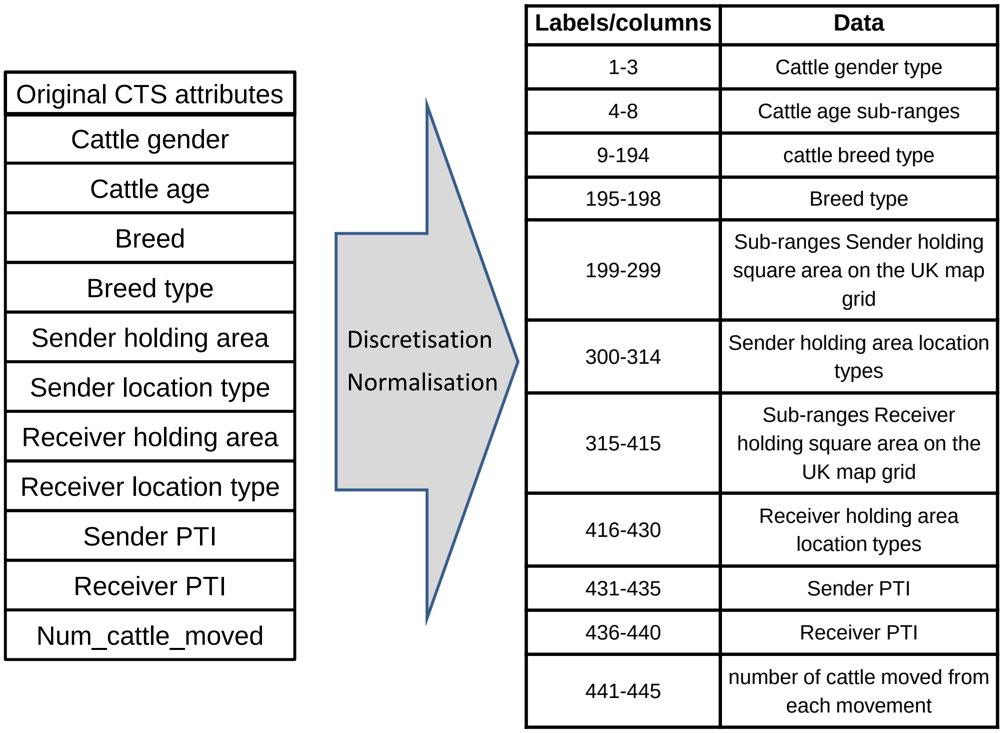
Table 3.1 presents an example database schema. In the example the data attributes are discretised and normalised according to their data types. The example considers a dataset comprising three attributes: (i) easting, (ii) number of cattle moved and (iii) gender. The first two are continuous attributes while the third is a nominal attribute. In this case, after the discretisation and normalisation process, the datasets will have been divided into 17 "column" attributes: (i) 10 columns ($\{1 \ldots 10\}$) representing the easting values, (ii) 5 columns ($\{11 \ldots 15\}$) representing the number of cattle moved

Attribute data type	Attribute	Range.
Continuous (double)	Easting	The values can be $0 - 100,000$ thus it can be divided into 10 sub-ranges, $0.0 \leq n \leq 100,000.0$, $100,000.0 \leq n \leq 200,000.0$, $200,000.0 \leq n \leq 300,000.0 \ \ldots \ n \geq 1,000,000.0$
Integer	Number of cattle moved	The values can be divided into five sub-ranges, $0<n<5$, $5<n<10$, $10<n<20$, $20<n<30$ and $n>30$
Nominal	Gender	The values are converted into male $= 1$ and female $= 2$

Table 3.1: Examples of discretisation and normalisation of data attributes

and (iii) 2 columns ($\{16 \ldots 17\}$) representing the two possible values for the gender attribute.

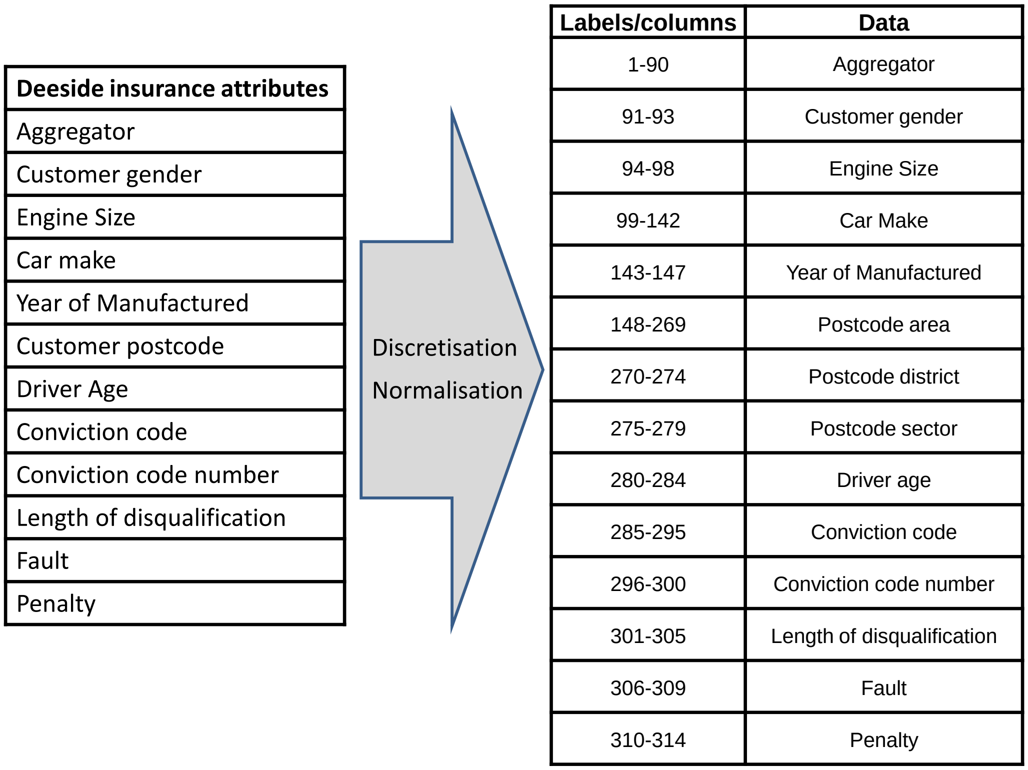
Figures 3.3, 3.4 and 3.5 summarise the discretisation and normalisation conducted with respect to the identified datasets. In each figure the original attributes are presented on the left and the derived attributes on the right. As a result, the CTS dataset has 445 attributes, the Deeside Insurance quotation dataset has 314 attributes and MAF Logistic Cargo dataset has 201 attributes.



Original CTS attributes
Cattle gender
Cattle age
Breed
Breed type
Sender holding area
Sender location type
Receiver holding area
Receiver location type
Sender PTI
Receiver PTI
Num_cattle_moved

Labels/columns	Data
1-3	Cattle gender type
4-8	Cattle age sub-ranges
9-194	cattle breed type
195-198	Breed type
199-299	Sub-ranges Sender holding square area on the UK map grid
300-314	Sender holding area location types
315-415	Sub-ranges Receiver holding square area on the UK map grid
416-430	Receiver holding area location types
431-435	Sender PTI
436-440	Receiver PTI
441-445	number of cattle moved from each movement

Figure 3.3: Discretisation and normalisation CTS attributes



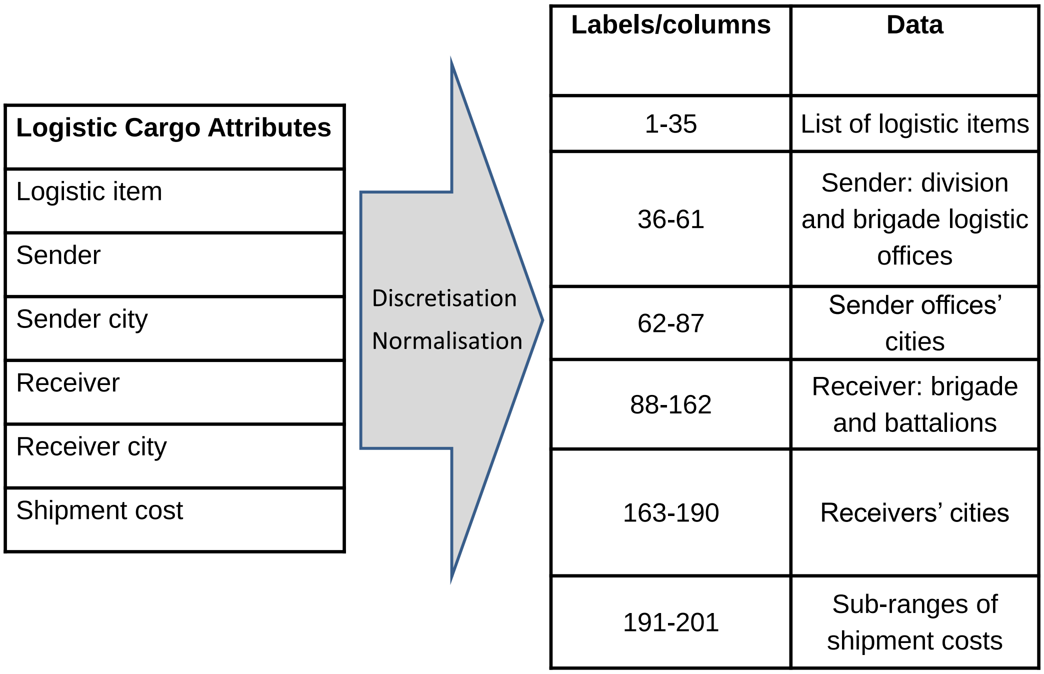
Deeside insurance attributes
Aggregator
Customer gender
Engine Size
Car make
Year of Manufactured
Customer postcode
Driver Age
Conviction code
Conviction code number
Length of disqualification
Fault
Penalty

Labels/columns	Data
1-90	Aggregator
91-93	Customer gender
94-98	Engine Size
99-142	Car Make
143-147	Year of Manufactured
148-269	Postcode area
270-274	Postcode district
275-279	Postcode sector
280-284	Driver age
285-295	Conviction code
296-300	Conviction code number
301-305	Length of disqualification
306-309	Fault
310-314	Penalty

Figure 3.4: Discretisation and normalisation Deeside Insurance quotation attributes

3.5 Summary

This chapter has described the datasets used for evaluating the proposed Frequent Pattern Trend Analysis and Prediction Modeling. Three datasets were selected for the evaluation: (i) the CTS database, (ii) the Deeside Insurance quotation database, and (iii) the MAF Logistic Cargo distribution database. An appropriate social network dataset was extracted from each of these encompassing a sequence of time stamps and episodes. In each case the dataset attributes were discretised and normalised to form a $\{1 \ldots N\}$ label/column binary valued dataset. The next chapter describes the proposed Frequent Pattern Trend Analysis mechanisms.



Logistic Cargo Attributes
Logistic item
Sender
Sender city
Receiver
Receiver city
Shipment cost

Labels/columns	Data
1-35	List of logistic items
36-61	Sender: division and brigade logistic offices
62-87	Sender offices' cities
88-162	Receiver: brigade and battalions
163-190	Receivers' cities
191-201	Sub-ranges of shipment costs

Figure 3.5: Discretisation and normalisation MAF Logistic Cargo distribution attributes

Chapter 4

The Frequent Pattern Trend Analysis

As noted in Chapter 1 the motivation for the research described in this thesis is the requirement for techniques and mechanisms to support the identification and analysis of trend information contained in network data so as to provide support for decision and policy makers. The objective is to investigate and evaluate mechanisms to identify dynamic changes in network data which, in turn, can be used to direct actions. This chapter describes the first part of the proposed Predictive Trend Mining Framework (PTMF), comprising the Frequent Pattern Trend Analysis (FPTA) modules, to support the analysis of temporal frequent patterns and trends in the context of Social Networks. The FPTA modules were designed to identify frequent patterns from a sequence of time stamped datasets so that the sequence of frequency counts (support values) associated with a particular pattern described a "trend line". As such the modules provide support for the analysis of time stamped datasets where the data is organised into episodes (each episode described by a sequence of time stamps). Referring back to the research objectives (Section 1.2) the proposed modules in this chapter are intended to provide effective mechanisms to: (i) discover temporal frequent patterns and trends in network data, and (ii) facilitate the analysis of these trends and patterns so as to identify behaviours that might exist across networks.

The FPTA element of PTMF comprises four modules: (i) Trend Identification, (ii) Trend Grouping, (iii) Pattern Migration Clustering and (iv) Pattern Migration Visualization. The Trend Identification module is used to identify a set of temporal patterns (trend lines) describing the fluctuating levels of support for individual patterns. The Trend Grouping module is then used to group the discovered trends using a Self Organising Map (SOM) [73]. The Pattern Migration Clustering module may be applied to identify how particular patterns "migrate" (or do not migrate) from one SOM node to another over one or more successive SOM maps. The Pattern Migration Clustering module is also used to cluster the identified pattern migrations, the idea here is that such migrations may provide interesting information in terms of temporal changes and communities of migrating patterns. To support end user interpretation of the output, the Pattern Migration Visualisation module provides a graphical representation of the migration results.

The rest of this chapter is organized as follows. Firstly, Section 4.1 presents a formal definition of the frequent trend mining problem as conceptualized with respect to this thesis. Section 4.2 introduces the FPTA modules in more detail. Each of the four modules is then described in the following four sections. Section 4.3 considers the algorithm for generating temporal patterns and trends. Section 4.4 discusses the process for grouping trends to discover "types" of trends; followed, in Section 4.5, by a description of the proposed technique for detecting pattern migrations and "communities" of migrating patterns. Section 4.6 then discusses the Pattern Migration Visualisation module. In each of these sections pseudo code is included (where appropriate) so as to explain the operation of each module. A discussion and some assumptions applied during the development of the modules are considered in Section 4.7. Finally, in Section 4.8, the chapter is concluded with a brief summary.

4.1 Formalism and Definitions

The input to the FPTA modules comprises a sequence of n time stamped datasets, $D = \{d_1, d_2, \ldots d_n\}$. To identify changes in trends (or lack of them), the available time stamps were subdivided into e episodes, each of equal length m, thus $n = e \times m$. The size of m, and hence the number of episodes e, will be application dependent. However, with respect to the selected datasets, a granularity of one month was used for the time stamps and hence m was set to 12; consequently each episode represented a year (for example, the GB cattle movement data was divided into four episodes: 2003, 2004, 2005 and 2006).

Each dataset comprises a binary valued table such that each record represents the traffic between a social network node pair in a given network. The level of detail provided will vary between applications; nodes may be described in terms of a single attribute or a number of attributes. Nodes may include information about the entity they represent, such as location information (for example post codes, or eastings and northings) and/or the nature of the attribute. For the GB cattle movement dataset, a number of node attributes were identified, such as: node type (farms, markets, abattoirs, etc.), address and location grid coordinates (eastings and northings). The quantity of traffic was defined in terms of a sequence of ranges. Additional traffic information may also be provided, for example in the case of the GB cattle movement application information concerning the nature of the cattle moved is included (breed type, gender, etc.). Thus, each record, in each dataset d_i is comprised of a subset of binary valued attributes taken from a global set of attributes $A = \{a_1, a_2, \ldots a_m\}$. Note that the number of records in each dataset need not be constant across the collection.

As already noted, in this thesis, a trend line t associated with a pattern I is defined in terms of the frequency of occurrence of I within D, over the sequence of time stamps within an episode. Thus, a trend line t, for a particular pattern I, comprises a set of values $t_I = \{v_v, v_2, \ldots v_m\}$ where each value represents an occurrence count of the pattern at a particular time stamp within an episode of length m. The collection of trend lines for a pattern I, T_I, is then comprised of a sequence of trend lines (one per episode) $\{t_{I_1}, t_{I_2}, \ldots t_{I_e}\}$ (where e is the number of episodes).

The entire collection of trends within a system is given by τ. The trend lines associated with a time stamp i is given by τ_i. Thus $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$. The entire collection of trends associated with a particular episode i is given by τe_i. Thus $T = \{\tau e_1, \tau e_2, \ldots, \tau e_e\}$. The trend lines associated with a specific episode e_i are given by $\tau e_i = \{\tau_k, \tau_{k+1}, \ldots, \tau_{k+m}\}$, where k is the first time stamp in the episode e_i.

4.2 Frequent Pattern Trend Analysis Modules

As noted above the proposed FPTA process involves four modules:

1. **Trend Identification**: The trend identification module comprises the mining unit for identifying and generating the frequent patterns and trends from the raw data.

2. **Trend Grouping**: The trend grouping module groups similar trends, using SOM technology, into trend clusters (each represented by a SOM node) one SOM per episode. Note that each cluster (SOM node) maintains the details of the frequent patterns that it represents.

3. **Pattern Migration Clustering**: The pattern migration clustering module groups pattern that migrate across the SOM network in a similar way. The aim is to identify “communities” whose associated trends change in a similar manner (or do not change) between episodes.

4. **Pattern Migration Visualisation**: The pattern migration visualization module incorporates a tool to illustrate the migration activities between trend clusters (SOM nodes) identified in module 3.

The FPTA process is shown in diagrammatic form in Figure 4.1. The process commences in the top-left corner of the figure with the data (a collection of networks). The input data is then processed, using the four modules, to identify and analyse the frequent patterns and trends, and give the final visualization (bottom-left corner of the figure). Each module is described in further detail in the following sub-sections; pseudo code and worked examples are included where appropriate. The nature of the software associated with each module is indicated (in Figure 4.1) in parenthesis.

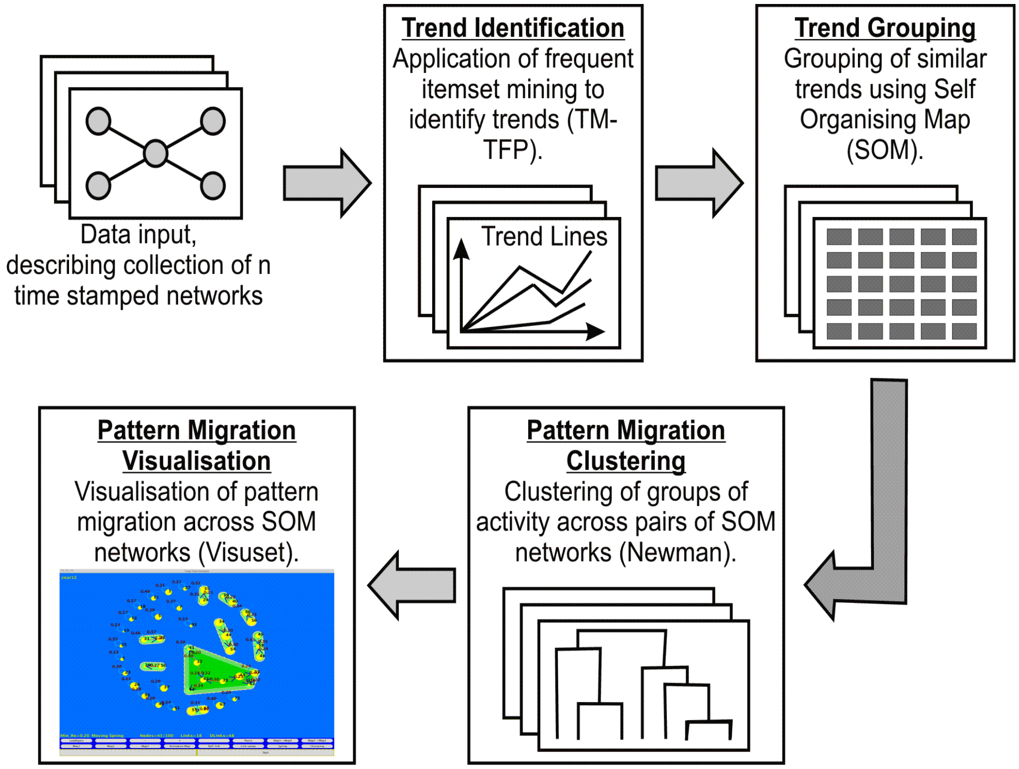

Data input,
describing collection of n
time stamped networks
Trend Identification
Application of frequent itemset mining to identify trends (TM-TFP).
Trend Lines
Trend Grouping
Grouping of similar trends using Self Organising Map (SOM).
Pattern Migration Clustering
Clustering of groups of activity across pairs of SOM networks (Newman).
Pattern Migration Visualisation
Visualisation of pattern migration across SOM networks (Visuset).

Figure 4.1: Schematic illustrating the operation of the FPTA

4.3 Trend Identification

As already noted the patterns of interest are frequent itemsets as popularised in Association Rule Mining [7]. To mine patterns and trends an extended version of the Total From Partial (TFP) algorithm [29, 30], described in Section 2.2.2, was incorporated into the Trend Identification module. TFP was selected because it is an established frequent pattern mining algorithm that has been shown to be efficient. An alternative might have been FP-growth [50] which was described in Section 2.2.2. As noted in Chapter 2, TFP is distinguished by its use of two data structures: (i) a P-tree used to both encapsulate the input data and record a partial frequency count for each pattern, and (ii) a T-tree to store the identified patterns together with their total frequency counts. Recall that the T-tree is essentially a reverse *set enumeration tree* that allows fast *look up*. Recall also that the T-tree comprises an array and node structure, each level in the T-tree is represented by an array (array indexes indicate item identifies) comprised of pointers (references) to node objects. The TFP algorithm follows an *apriori* style of operation to generate frequent items sets whereby the *antimonotone property* of item sets is used to limit the search space. The well documented *support framework* is used, whereby a frequency count threshold (the *support threshold*) is used to define "interesting" patterns; typically the lower the support threshold the more patterns that are discovered.

The TFP algorithm, in its original form, was not designed to address the temporal aspect of frequent pattern mining. For the purpose of the FPTA process, the TFP algorithm was therefore modified and extended so that sequences of datasets could be processed, and the discovered frequent patterns stored, in a way that would allow for differentiation between individual time stamps and episodes. The resulting algorithm was called TM-TFP (Trend Mining-TFP) which incorporated a TM-tree to store the desired patterns. Further details concerning the TM-TFP algorithm, will be described in the following sub-section (note that similar descriptions have been published previously by the author in [96] and [97]). The output from the TM-TFP algorithm is the desired collection of trends $T = \{T_1, T_2, \ldots, T_e\}$. Experiments using a variety of network datasets (reported in [95]) have indicated that a large number of trends are often identified. More details concerning these experiments are presented in the Chapter 5. Of course, the number of patterns to be considered can be reduced by using a higher support threshold, but the established argument against this expedient is that potential interesting patterns may be overlooked. In situations where a pattern is sometimes frequent (above or equal to the support threshold) and sometimes infrequent (below the support threshold), a value of 0 is recorded where the support count falls below the threshold. It can be argued that where itemsets are sometimes frequent and sometimes not frequent the support count should always be recorded, however this was found to introduce an unacceptable computational overhead. The author did conduct some experiments using a negative border, as advocated in [120], however this still resulted in some frequent sets falling below the negative border threshold. Therefore the straightforward expedient of using relatively low support thresholds and ignoring the frequency counts for infrequent patterns (replacing it with a count of 0) was adopted. Note also that where a pattern is not supported with respect to an episode, no trend line is generated even if the pattern is supported in some other episode.

4.3.1 Trend Mining-TFP Algorithm

In this sub-section the TM-TFP algorithm, which is directed at identifying sequences of frequent patterns (itemsets) within time stamped data is described. TM-TFP utilizes the T-tree and P-tree data structures for storing and indexing the itemsets. In Chapter 2, an example was given of how patterns and support values are stored using P-trees and T-trees. The TM-TFP algorithm is founded on the TFP algorithm. The distinction between TFP and TM-TFP is that the latter uses an additional TM-tree to integrate T-trees associated with individual time stamps. The TM-tree has a similar structure to T-trees. Each node in a TM-tree has two fields: (i) trend and (ii) reference. The first field consists of a vector of support values that describes the associated trend line. The second holds a reference (pointer) to the next level of the TM-tree. As in the case of the

T-tree, the individual items making up a frequent pattern are not explicitly stored as this can be obtained simply from the tree structure. The TM-tree also has a TM-tree header which holds references to a set of n T-trees (recall that n is the number of time stamps in a given system). Figure 4.2 shows the structure of the TM-tree. For ease of reading the TM-Tree presented in the figure only holds trends of length two $[v_{n_i}, v_{n_j}]$.

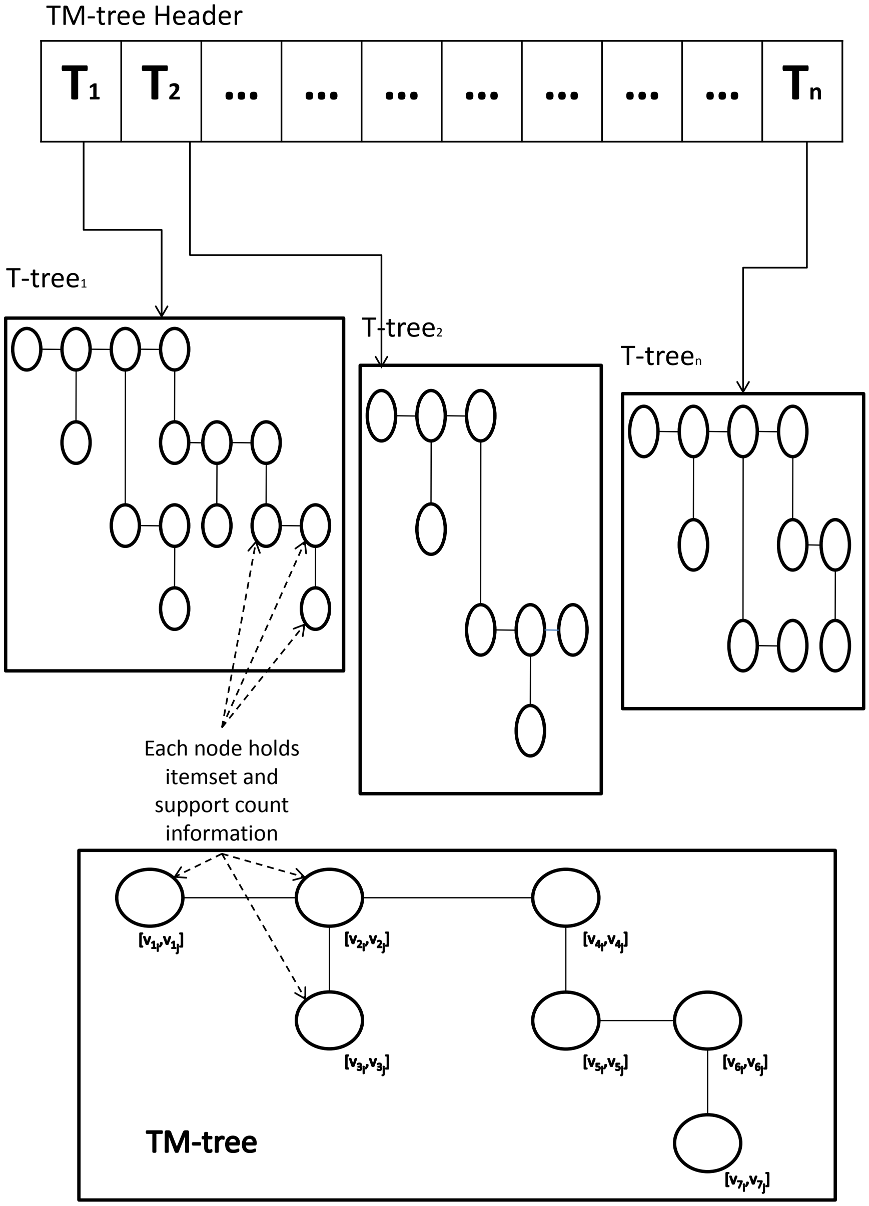


Figure 4.2: Structure of TM-tree

Algorithm 4.1: TM-TFP algorithm for building TM-tree

```
  input  : A set of n datasets, $D = \{d_1, d_2, \ldots d_n\}$, minSupport
  output: TM-tree holding frequent itemsets' trends in $D$
  //  Initialise TM-Tree header file
1 TM-tree Header size ← $n$;
2 for $i \leftarrow 0$ to (n-1) do
3     Create P-tree$_i$ from $d_i$;
4     Create T-tree$_i$, using minSupport;
5     TM-tree Header [$i$] ← T-tree$_i$ reference;
6 end
7 buildTM_Ttree() //  Algortihm 4.2
```

An overview of the operation of the TM-TFP algorithm is given by the pseudo code presented in Algorithm 4.1. The algorithm commences by initialising the TM-tree header according to the number of time stamped datasets held in D (line 1). Then, lines 2 to 5, it loops through the datasets d_1 to d_n contained in D and creates individual P-trees and T-trees, one per dataset d_i, using the TFP algorithm. Note that a reference to each generated T-tree is stored in the TM-tree header (line 5). Finally, line 7, the `buildTM_Ttree()` method is called to process the collection of T-trees built from $\{d_1, d_2, \ldots d_n\}$, and construct the desired TM-tree.

Algorithm 4.2: buildTM_Ttree()

```
  input  : A set of n T-trees
  output: TM-tree holding frequent itemsets in $D$
  //  builds TM-tree by processing a collection of T-trees built for
      $D_n$
1 Initialise TM-tree to accommodate all T-trees from 0 to $n - 1$;
2 for $i \leftarrow 0$ to (n-1) do
3     if T-tree$_i \neq \emptyset$ then
4         buildTM_Ttree (T-tree$_i$,i) // Algorithm 4.3
5     end
6 end
```

Algorithm 4.2 describes the `buildTM_Ttree()` method. The algorithm commences (line 1) by determining the total number of frequent itemsets that are required to be held in the TM-trees. The TM-tree is then initialised using this number so that it can hold all combinations of frequent itemsets and trends contained in the n T-trees. The TM-tree is then constructed (line 2 to 4) by repeated calls to the `buildTM_Ttree` method. The pseudo code for this method is presented in Algorithm 4.3. The input to this algorithm is the current T-tree, T-tree$_i$, and the current time stamp i. Note that on line 2 of Algorithm 4.2 there is a test for the empty T-tree situation (which may exist if a very high support threshold is used). If the input T-tree is not empty Algorithm 4.3 proceeds by processing the top level of the given T-tree (T-Tree$_i$). For each element in this top level, T-tree$_i[k]$, if the element is not empty the algorithm calls (line 3) the `addToTMtree` method (Algorithm 4.5) with the associated support. If the T-tree$_i[k]$ has child nodes, these are then processed (Algorithm 4.4). If the element is empty (not supported) the algorithm calls (line 8) the `addToTMtree` method with a support value of 0.

Algorithm 4.3: buildTM_Ttree(T-tree$_i$,i)

```
   input  : T-tree_i, time stamp i
   output: TM-tree holding frequent itemsets in D
   //  loop through top level of current T-tree
 1 for $k \leftarrow 1$ to numOfoneItemsets do
 2     if T-tree_i[k] $\neq \emptyset$ then
 3         addToTMtree (T-tree_i[k].support, i) //  Algorithm 4.5
           //  move down a level
 4         if T-tree_i[k] has child node then
 5             buildTM_Ttree (T-tree_i[k].child, i,k-1) //  Algorithm 4.4
 6         end
 7     else
 8         addToTMtree (0, i)
 9     end
10 end
```

Algorithm 4.4: buildTM_Ttree(T-tree$_i$,i,size)

```
   input  : $T - tree_i$, time stamp i, size
   output: TM-tree holding frequent itemsets in D
 1 for $k \leftarrow 1$ to size (of current node level of the current T-tree) do
 2     if  T-tree_i[k] $\neq \emptyset$ then
 3         addToTMtree (T-tree_i[k].support, i) //  Algorithm 4.5
           //  move down a level
 4         if T-tree_i[k] has child node then
 5             buildTM_Ttree (T-tree_i[k].child,i,k-1) //  Algorithm 4.4
 6         end
 7     else
 8         addToTMtree (0, i)
 9     end
10 end
```

Algorithm 4.4 processes the child nodes of a given T-tree$_i$. The size parameter is the number of elements at the current node in the current branch of the given T-tree. The algorithm loops through the elements in the current level of the tree (line 1). If the node represented by an element is not empty (i.e. it represents a supported item) the equivalent node in the TM tree is updated with the support count (Algorithm 4.5). In line 4 the algorithm tests whether the current node (element) has a child branch associated with it. If so this next level is processed by a repeat call to Algorithm 4.4, and so on until there are no more child branches to be processed. The process of building the TM-tree continues in this manner until all T-trees have been considered. Line 8 deals with the situation when a T-tree level element is not supported, in this case a call is made to the `addToTMtree` method with a support value of 0.

Algorithm 4.5: addToTMtree(support, i)

```
input  : support, timestamp $i$
output : TM-tree updated with itemset support
TM-tree.trend[$i$] = T-tree$_i$.support;
return;
```

Algorithm 4.5 adds the support value of an itemset in a given T-tree$_i$ to the TM-tree. If the given T-tree$_i$ element (node) representing the itemset is empty (the itemset is not supported) the support value will be 0. Recall that zero is used as a flag to indicate that an itemset is not supported. Whatever the case the added support for the itemset will form part of the eventual trend line, stored at the TM-tree node, for the itemset.

Figure 4.3 presents a worked example of the building of a TM-tree. Let $d_1 = \{ab, acd, ab\}$ and $d_2 = \{bd, cd, abd\}$, the TM-TFP algorithm starts by creating P-tree$_1$ and T-tree$_1$ for d_1, followed by P-tree$_2$ and T-tree$_2$ for d_2. Each T-tree is linked to the TM-tree header. Subsequently, the algorithm scans through T-tree$_1$ and T-tree$_2$ to build the TM-tree with a collection support values from both T-trees to form trends.

A representation of the content of part of a TM-tree is given in Figure 4.4 (generated using an output facility included in the TM-TFP algorithm for diagnostic purposes). From the figure it can be seen that a TM-tree node holds three pieces of information: (i) the itemset identifier (held implicitly), (ii) a trend represented by a set of support counts and (iii) a reference to the next level in the TM-tree.

Table 4.1 presents the number of patterns discovered from the GB cattle movement dataset using three different support thresholds (the first column gives the episode identifier). The table serves to demonstrate that a large number of trends are discovered using TM-TFP. This was one of motivations for the inclusion of the Trend Grouping module into the process of FPTA. The trend grouping module was also motivated by a desire to formulate a mechanism to support the analysis of the discovered trends. The module is discussed further in the following sub-section.

4.4 Trend Grouping

As indicated in Table 4.1, and as noted at the end of the previous section, a large number of trends are typically identified using TM-TFP. A proposed mechanism, to support the desired trend analysis, incorporated into the process of FPTA, is to group the discovered trends according to their distinguishing features (increasing, decreasing, steady and so on). This section begins, Sub-section 4.4.1, by describing the proposed

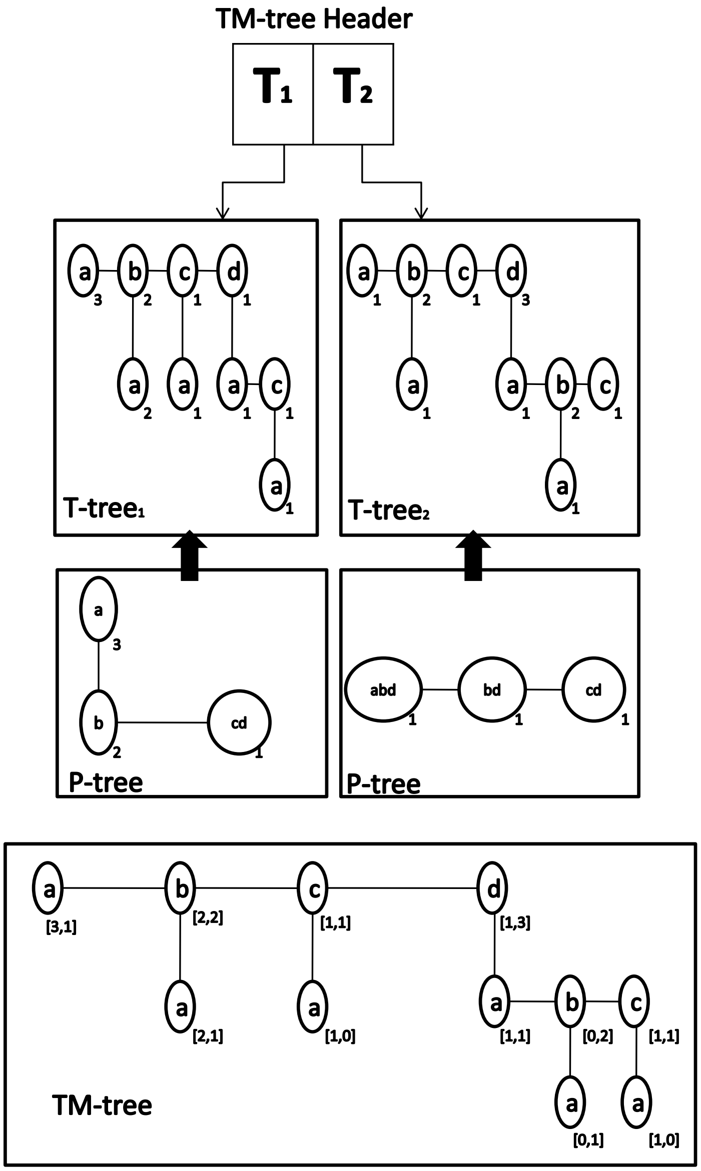
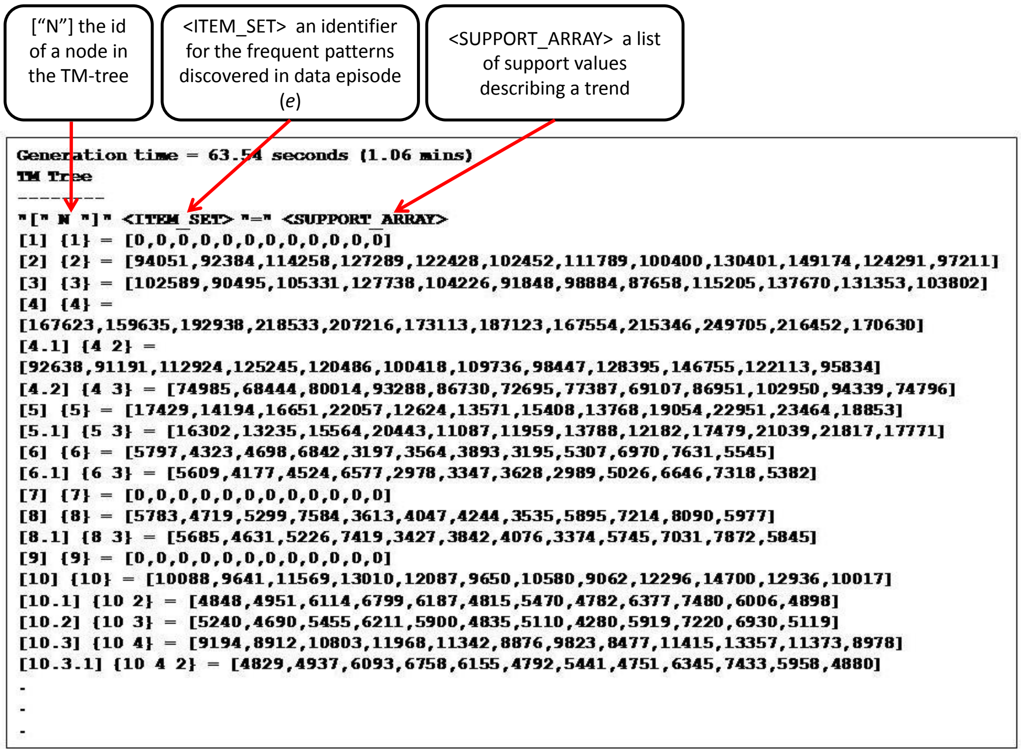


Figure 4.3: A worked example of the operation of the TM-TFP Algorithm

process. The process starts by grouping similar trends using a SOM. The intuition here was that end users were expected to be interested in particular types of trends. Using a SOM, similar trends can be clustered at particular nodes in a SOM map. A separate SOM map can be generated for each episode, and thus the maps can be viewed in terms of a sequence of maps (with respect to the applications used to evaluate the proposed FPTA modules each map represented a year of activity). Prior to starting the SOM process the SOM map must be initialized. A discussion on the optimum size for a SOM map is therefore presented in Sub-section 4.4.2.



["N"] the id of a node in the TM-tree

<ITEM_SET> an identifier for the frequent patterns discovered in data episode (*e*)

<SUPPORT_ARRAY> a list of support values describing a trend

```
Generation time = 63.54 seconds (1.06 mins)
TM Tree
---------
"[" N "]" <ITEM_SET> "=" <SUPPORT_ARRAY>
[1] {1} = [0,0,0,0,0,0,0,0,0,0,0,0]
[2] {2} = [94051,92384,114258,127289,122428,102452,111789,100400,130401,149174,124291,97211]
[3] {3} = [102589,90495,105331,127738,104226,91848,98884,87658,115205,137670,131353,103802]
[4] {4} =
[167623,159635,192938,218533,207216,173113,187123,167554,215346,249705,216452,170630]
[4.1] {4 2} =
[92638,91191,112924,125245,120486,100418,109736,98447,128395,146755,122113,95834]
[4.2] {4 3} = [74985,68444,80014,93288,86730,72695,77387,69107,86951,102950,94339,74796]
[5] {5} = [17429,14194,16651,22057,12624,13571,15408,13768,19054,22951,23464,18853]
[5.1] {5 3} = [16302,13235,15564,20443,11087,11959,13788,12182,17479,21039,21817,17771]
[6] {6} = [5797,4323,4698,6842,3197,3564,3893,3195,5307,6970,7631,5545]
[6.1] {6 3} = [5609,4177,4524,6577,2978,3347,3628,2989,5026,6646,7318,5382]
[7] {7} = [0,0,0,0,0,0,0,0,0,0,0,0]
[8] {8} = [5783,4719,5299,7584,3613,4047,4244,3535,5895,7214,8090,5977]
[8.1] {8 3} = [5685,4631,5226,7419,3427,3842,4076,3374,5745,7031,7872,5845]
[9] {9} = [0,0,0,0,0,0,0,0,0,0,0,0]
[10] {10} = [10088,9641,11569,13010,12087,9650,10580,9062,12296,14700,12936,10017]
[10.1] {10 2} = [4848,4951,6114,6799,6187,4815,5470,4782,6377,7480,6006,4898]
[10.2] {10 3} = [5240,4690,5455,6211,5900,4835,5110,4280,5919,7220,6930,5119]
[10.3] {10 4} = [9194,8912,10803,11968,11342,8876,9823,8477,11415,13357,11373,8978]
[10.3.1] {10 4 2} = [4829,4937,6093,6758,6155,4792,5441,4751,6345,7433,5958,4880]
-
-
-
```

Figure 4.4: An example of the diagnostic output from the TM-TFP algorithm comprising a list of frequent patterns and 12 months of trend values

Episode	Support Threshold		
(year)	0.5%	0.8%	1.00%
2003	63,117	34,858	25,738
2004	66,870	36,489	27,055
2005	65,154	35,626	25,954
2006	62,713	33,795	24,740

Table 4.1: Number of trends identified using TM-TFP for a sequence of four GB Cattle Movement network episodes and a range of support thresholds

4.4.1 Trend Clustering using Self Organizing Maps

To group the trends one SOM was created per data episode. SOMs [73] may be viewed as a type of feed-forward, back propagation, neural network that comprises an input layer and an output layer (an $i \times j$ grid). The cells in the $i \times j$ grid are referred to as nodes; each node potentially represents a trend cluster (a grouping of trends that display similar geometry). Recall that in the work described in this thesis, the input layer comprises the trends (trend lines formed of n support counts associated with each frequent pattern) and the output layer the trend clusters. Each output node (map node) in the output layer is connected to every input node in the input layer, a trend line, which is assigned a set of weight vectors, w. The dimension of the weight vectors is the same as the dimension of the trend lines of interest, for example in this thesis trend lines are of length 12 (months). The SOM was then "trained" using a training input dataset. Algorithm 4.6 provides the trend grouping pseudo code for clustering the trend lines generated using TM-TFP. With reference to this pseudo code the SOM is first initialised (line 1) with a predefined $x \times y$ grid (map). A discussion on the optimum size of a grid/map is presented in Sub-section 4.4.2 below.

Algorithm 4.6: Trend Grouping using SOM

```
   input  : $T = \{\tau e_1, \tau e_2, \ldots, \tau e_e\}$
   output: SOM prototype map and $n$ trend line maps
   //  generate a SOM prototype map
1  Initialise a SOM prototype map with $x \times y$ nodes;
2  Assign weight vectors, w, to the map nodes;
3  for $i \leftarrow 0$ to $|\tau_{e_1}|$ do
4      Find the "winning" node for trend line $t_{1_i}$ in the prototype map;
5      Adjust the weight vectors of nearby map nodes accordingly;
6  end
   //  generate a SOM trend line maps
7  for $k \leftarrow 0$ to $e$ do
8      Initialise a SOM trend line map, with $x \times y$ nodes  for episode k;
9      for $i \leftarrow 0$ to $|\tau_{e_k}|$ do
10         Plot $t_{k_i}$ onto the prototype map for episode k;
11     end
12 end
```

The SOM was thus trained using the trend lines associated with the frequent patterns discovered in the first data episode (e_1) (line 3 to 5). Each record in τe_1 (defined in Section 4.1) was presented to the SOM in turn. The output nodes then "compete" for each record. Once a record has been assigned to the "winning" map node, the network's weightings are adjusted to reflect this new position. At first the adjustments are relatively large, but as the training continues the adjustments become smaller and smaller. A distance function[1] and a neighbourhood function[2] were used to determine similarity. A feature of the adjustment was that adjacent nodes would come to hold similar records; the greatest dissimilarity would be between nodes at opposite corners of the map. At the end of the SOM training phase, a prototype map was produced that represented the types of trend lines that existed within the set of identified trend lines in τe_1. Copies of the resulting *prototype map* were then populated with data from all e episodes (τe_1 to τe_e), to produce a sequence of e maps $M = \{M_1, M_2, \ldots, M_e\}$ (line 8 to 12). Using this SOM based clustering process the substantial number of trends that are typically identified using TM-TFP could be grouped according to their trend

[1] A Euclidean function was adopted with respect to the work described in this thesis.

[2] Gaussian function was used to determine the neighbourhood size of the map.

"types" so as to consequently aid analysis. Figure 4.5 illustrates the process. The figure features four episodes which are used to generate four SOM maps (labeled I, II, II, IV) based on the prototype map.

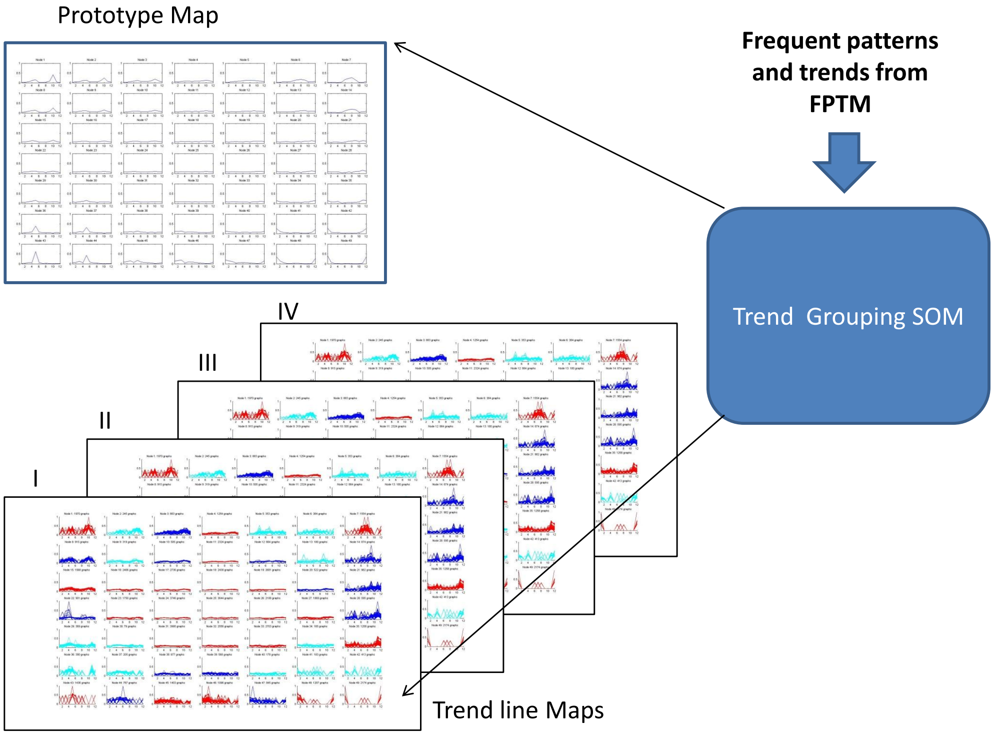


Figure 4.5: SOM Prototype and Trend lines maps

The author experimented with a number of different mechanisms for training the SOM, including: (i) devising specific trends to be represented by individual nodes, (ii) generating a collection of all the mathematically possible trends and training the SOM using this set, and (iii) using some or all of the trends in the first epoch to be considered. The first required prior knowledge of the trend configurations of interest; which, it was conjectured, tended to defeat the objective of the trend mining process. The second mechanism, it was discovered, resulted in maps for which the majority of nodes were empty. The third option was therefore adopted. The third mechanism also supported the idea of identifying changes in trends associated with particular frequent patterns between episodes.

4.4.2 Discussion on SOM node configuration

As noted in Chapter 2, it is difficult to predefine the optimal number of SOM map clusters. A set of experiments was conducted to determine the most appropriate configuration of SOM nodes. The experiments were conducted using the GB cattle movement data since it features the largest collection of patterns and trends when compared to Deeside Insurance Quote and Malaysian Armed Forces Logistic Cargo networks. A very large grid size would allow for the grouping of trend lines into a greater number (possibly more accurate) map nodes (clusters), however this would also result in an undesirable computational overhead and in many cases might not serve to resolve the situation as many of the map nodes may remain empty (i.e. the items are consistently held in a small number of map nodes such that increasing the size of i and j has little or no effect). Figures 4.6, 4.7 and 4.8 present prototype maps of the 2003 cattle movement dataset using three sizes of map 7×7, 10×10 and 12×12 respectively. Inspection of these prototype maps indicates that each produced similar trend clusters or sub-clusters however it is apparent that the bigger map sizes features a distribution of the data that is more "distinct" or "finer". The trend line maps shown in Figure 4.9, 4.10 and 4.11, which were generated using the prototype maps in Figures 4.6, 4.7 and 4.8 respectively, demonstrate that the larger the SOM grid, the greater the possibility of having empty map nodes. There are no empty map nodes in the SOM given in Figure 4.9. In Figure 4.10, there is one empty map node, node 92. In Figure 4.11 there are four empty map nodes, nodes 121, 122, 134 and 143. Thus, in the case of the GB cattle movement (network) a 10×10 node SOM was considered to be the most effective as this gave a good decomposition while still ensuring computational tractability. For the insurance quotation and logistic cargo distribution datasets, a 7×7 node SOM was found to be more suitable.

4.5 Pattern Migration Clustering

The third module in the proposed FTPA process provides for further analysis of the patterns and trends contained in the generated SOMs (one per episode) with respect to the concept of pattern migration. The motivation here was that, at least in the context of the networks used for evaluation purposes in this thesis, discussion with potential end users indicated that they would be interested in how trends associated with particular patterns changed over time. In other words, in the context of the SOM maps, how the trend lines associated with particular patterns migrate (or did not migrate) across a sequence of maps. For this purpose each sequential pair of SOM maps was used to construct a second *migration* network/map comprising, $i \times j$ nodes and potentially $(i \times j)^2$ links (including "self links").

Given two SOM maps M_e and M_{e+1}, the *from map* and the *to map* respectively, the nodes in a migration network were labeled with the number of patterns held in the node in map M_e (i.e. the *from map*). The links then represented the migration of patterns from M_e to M_{e+1}, and were labeled with the number of migrating patterns (thus a "traffic" value). The higher the value the stronger the link. The process of visualising such migration networks is discussed in the following sections (Section 4.6). The remainder of this section is organised as follows. Sub-section 4.5.1 describes the process of detecting migrations of frequent patterns between two SOM maps M_e and M_{e+1}. This is followed in Sub-section 4.5.2 by a description of the clustering of the identified pattern migrations; a hierarchical clustering method is suggested. A worked example of the clustering of the identified pattern migrations, using the Newman method, is shown in Sub-section 4.5.3.

4.5.1 Pattern Migration

The changes in trends associated with patterns can be measured by interpreting the SOM trend line maps in terms of a rectangular (2-D plane) where each point in the plane represents a SOM node. A Manhattan or Euclidean distance function can then be applied to determine the distance "traveled" by the patterns between nodes in successive SOM maps, which in turn can be used to observe the similarities and differences between pattern trends across episodes. The greater the distance a pattern moves the more significant the change. Thus, given a sequence of trend line SOM maps, comparisons can be made to see how trends associated with individual frequent patterns change by analyzing the nodes in which they appear. The trend cluster analysis pseudo code is described in Algorithm 4.7.

Algorithm 4.7: Trend Cluster Analysis

```
   input  : FP = Set of all frequent patterns in episodes $\{e_1, e_2 \ldots, e_e\}$
   output: Sequence of $(e-1)$ Migration Matrices
1  Define Table measuring $\mid FP \mid \times e$;
2  for $i \leftarrow 1$ to $e$ (step through episodes) do
3      for $j \leftarrow 1$ to $\mid FP \mid$(step through the set of FP) do
4          Table[i][j] = Table[i][j] $\cup$ SOM node ID for $e_i$;
5      end
6  end
7  Define $(e-1)$ Migration Matrices (MMs), each measuring $(x \times x)$ where $x$ is the
   number of SOM nodes;
8  for $i \leftarrow 1$ to $(e-1)$ do
9      for $k \leftarrow 1$ to $\mid FP \mid$ do
10         Increment count at MM$_i$[Table[K][i]][Table[K][i+1]];
11     end
12 end
```

The algorithm commences by defining a $|FP| \times e$ table. The table is populated with the SOM node IDs, the frequent pattern trend cluster, for each discovered frequent pattern in $\{e_1, e_2, \ldots, e_e\}$ for SOM maps $M = \{M_1, M_2, \ldots, M_e\}$ (line 4). Then in line 7, the algorithm defines a sequence of $e-1$ Migration Matrices (MMs) for each pair of SOM M_e and M_{e+1}, each measuring $x \times x$. The process continues by comparing the node numbers of the frequent pattern and counting the pattern migrations for each node ID (trend cluster) between SOM M_e and M_{e+1} (lines 8 and 10).

Subsequently, the module also produces a trend cluster analysis of the pattern migrations between trend clusters. The analysis comprises a comparison of pattern migrations for each pair of SOM M_e and M_{e+1}. The number of patterns migrating from node$_i$ in M_e to node$_j$ in M_{e+1} are recorded. It is also possible to determine how the sizes of trend clusters in a given pair of SOMs, M_e and M_{e+1}, change. This analysis thus provides for identification of patterns that move, or do not move, between successive SOM nodes, which may be of interest given particular applications.

4.5.2 Pattern Migration Hierarchical Clustering

To aid the further analysis of the identified trend migrations it was also considered desirable to identify "communities" within networks, i.e. clusters of nodes which were "strongly" connected (feature significant migration). This would indicate significant groupings of patterns whose associated trend lines where changing between episodes e_k and e_{k+1}. An agglomerative hierarchical clustering mechanism, founded on the Newman method [91] for identifying clusters in network data, was therefore adopted. Newman proceeds in the standard iterative manner on which agglomerative hierarchical clustering algorithms are founded. The process starts with a number of clusters equivalent to the number of nodes[3]. The two clusters (nodes) with the greatest "similarity" are then combined to form a merged cluster. The process continues until a "best" cluster configuration is arrived at or all nodes are merged into a single cluster. The overall process is typically conceptualised in the form of a dendrogram. Best similarity is defined in terms of the *Q-value*, this is a "modularity" value which is calculated as follows:

$$Q_i = \sum_{i=1}^{i=n}(c_{ii} - a_i^2) \tag{4.1}$$

where Q_i is the Q-value associated with the *current* cluster i, n is the total number of nodes in the network, c_{ii} is the fraction of intra-cluster (within cluster) links in cluster i over the total number of links in the network, and a_i^2 is the fraction of links that end in the nodes in cluster i if the edges were attached at random. The value a_i is calculated as follows:

$$a_i = \sum_{j=1}^{j=n} c_{ji} \tag{4.2}$$

where c_{ij} is the fraction of inter-cluster links, between the current cluster i and the cluster j, over the total number of links in the network.

Thus on each iteration the Q-values for all possible cluster pairings are calculated and the pairing with the highest Q-value selected for merging. The process proceeds

[3] The alternative is divisive hierarchical clustering where we start with a single cluster.

i	c_{ii}	a_i	a_i^2	Q
A	0.00	0.10	0.01	-0.01
B	0.00	0.10	0.01	-0.01
C	0.00	0.20	0.04	-0.04
D	0.00	0.10	0.01	-0.01

Table 4.2: Start Condition

Groups			Internal Links									
1	2	3	c_{11}	c_{22}	c_{33}	a_1	a_2	a_3	a_1^2	a_2^2	a_3^2	Q
AB	C	D	0.40	0.00	0.00	0.40	0.40	0.20	0.16	0.16	0.04	0.04
AC	B	D	0.00	0.00	0.00	0.60	0.20	0.20	0.36	0.04	0.04	-0.44
AD	B	C	0.00	0.00	0.00	0.40	0.20	0.40	0.16	0.04	0.16	-0.36
BC	A	D	0.00	0.20	0.00	0.60	0.20	0.20	0.36	0.04	0.04	-0.24
BD	A	C	0.00	0.00	0.00	0.40	0.20	0.40	0.16	0.04	0.16	-0.36
CD	A	B	0.00	0.40	0.00	0.60	0.20	0.20	0.36	0.04	0.04	-0.04

Table 4.3: First Iteration

until a best cluster configuration is achieved. This is defined as the configuration with the highest overall Q-value. Generally speaking, if the Q-value is above 0.3 then communities can be said to exist within the target network; the value of 0.3 was derived experimentally by Newman and Girvan [92]. Note that if all nodes are placed in one group the Q-value will be 0.0 (i.e. a very poor clustering). A worked example is presented in the following subsection. The identified clustering (communities) are then displayed as “islands” in the following stage in the FPTA process. This will be described in Section 4.6.

4.5.3 Worked Example of Hierarchical Clustering Using Newman

Considering the simple example network presented in Figure 4.12. The Q value for this network at the start of the process, when each vertex is considered to represent a group, is (using data from Table 4.2):

$$Q = -0.01 - 0.01 - 0.04 - 0.01 = -0.07$$

There are six potential joins AB, AC, AD, BC, BD and CD; giving rise to six potential configurations. Calculating the Q-value for each configuration (Table 4.3) gives a best Q-value of 0.04, this therefore represents the first join and the configuration $\{AB, C, D\}$ is generated.

For the next join, there are three possible configurations: $\{ABC, D\}$, $\{ABD, C\}$ and $\{AB, CD\}$. Calculating the Q-value for each of these configurations (Table 4.3) gives a best Q-value of 0.28, so this is the second join and the configuration $\{AB, CD\}$ is formed.

Groups		Internal Links						
1	2	C_{11}	C_{22}	a_1	a_2	a_1^2	a_2^2	Q
ABC	D	0.60	0.00	0.80	0.20	0.64	0.04	-0.08
ABD	C	0.40	0.00	0.60	0.40	0.36	0.16	-0.12
AB	CD	0.40	0.40	0.40	0.60	0.16	0.36	0.28

Table 4.4: Second Iteration

For the third iteration the only remaining option is to combine all the vertices, this will give a Q-value of 0.0. The discovered maximal value for Q is then 0.28 and hence the configuration associated with this value, $\{AB, CD\}$, is selected as the best grouping (clustering). The dendrogram for the example is given in Figure 4.13.

4.6 Pattern Visualisation and Animation using Visuset

It is often said that SOMs are a visualization technique, reference to Figures 4.9, 4.10 and 4.11 supports this view. However, it was felt that a better form of visualisation was desirable, especially in the context of the migration maps identified above. The proposed Pattern Migration Visualisation module provides two forms of visualisation (founded on the Visuset software system [94]):

1. Visualisation of pattern migrations between two successive SOMs.
2. Animation of the pattern migrations between three successive SOMs.

In each case the visualisation (animation) includes the pattern migration communities discovered, using Newman, as described above. The communities are depicted as “islands” demarcated by a “shoreline” (for aesthetic purposes the islands are also contoured, although no meaning should be attached to these contours). The visualisation process is described in Sub-section 4.6.1, and the animation in Sub-section 4.6.2, below.

4.6.1 Visualisation of Pattern Migration

For the visualisation, Visuset locates nodes in a 2-D “drawing area” using the *Spring Model* [60]. The spring model for drawing graphs in 2-D space is designed to locate nodes in the space in a manner that is both aesthetically pleasing and limits the number of edges that cross over one another. The graph to be depicted is conceptualised in terms of a physical system where the edges represent springs and the nodes inanimate objects connected by the springs. Nodes connected by “strong springs” therefore attract one another while nodes connected by “weak springs” repulse one another. The graphs are drawn following an iterative process. Nodes are initially located within the 2-D space using some set of (random) default locations (usually defined in terms of an x and y coordinate system) and, as the process proceeds, pairs of nodes connected by strong springs are "pulled" together. In the context of FPTA, the spring value was defined in terms of a *correlation coefficient* (C):

$$C_{ij} = \frac{X}{\sqrt{(|M_{e_k i}| \times |M_{e_{k+1} j}|)}} \tag{4.3}$$

where C_{ij} is the correlation coefficient between a node i in SOM M_{e_k} and a node j in SOM $M_{e_{k+1}}$ (note that i and j can represent the same node but in two different maps), X is the number of patterns that have moved from node i to j and $|M_{e_k i}|$ ($|M_{e_{k+1} j}|$) is the number patterns at node i (j) in SOM M_{e_k} ($M_{e_{k+1}}$). A migration is considered "interesting", and thus highlighted, if C is above a specified minimum relationship threshold (Min-Rel). With respect to the GB cattle movement data network, a threshold of 0.2 was found to provide a good working Min-Rel value; although Visuset does allow users to specify, and experiment with whatever Min-Rel value they like. The Min-Rel value is also used to prune links and nodes; any link whose C-value is below the Min-Rel value is not depicted in the visualisation, similarly any node that has no links with a C-value above Min-Rel is not depicted.

The Visuset spring model algorithm (a simplified version) proceeds as follows:

1. Set drawing area size constants, $SIZEX$ and $SIZEY$.

2. For all pair of nodes, allocate an *ideal distance*, $IDIST_{ij}$, where i and j are node numbers. In the current implementation: if a pair has a link, the distance is set as 200 pixels; otherwise it is set to 500 pixels.

3. Set initial coordinates for all nodes. All nodes are "queued" in sequence, according to their node number, from the top-left of the drawing area to the bottom-right.

4. For all node pairs determine the actual pixel distance $RDIST_{ij}$ (where i and j are node numbers).

5. For all nodes, recalculate the coordinates using equations 4.4 and 4.5 where: $node_{i_x}$ ($node_{i_y}$) is the x (y) coordinate of $Node_i$, n is the number of nodes to be depicted, K is the *spring constant*, and dx_{ij} (dy_{ij}) is the absolute value of $Node_{i_x} - Node_{j_x}$ ($Node_{i_y} - Node_{j_y}$).

6. If $dx_{ij} + dy_{ij}$ is below a specified threshold (in terms of a number of pixels), or if some maximal number of iterations is reached, exit.

7. Go to Step 4.

$$node_{i_x} = node_{i_x} + \sum_{j=1}^{j=n} \left(dx_{ij} \times K \times \left(1 - \frac{IDIST_{ij}}{RDIST_{ij}} \right) \right) \tag{4.4}$$

$$node_{i_y} = node_{i_y} + \sum_{j=1}^{j=n} \left(dy_{ij} \times K \times \left(1 - \frac{IDIST_{ij}}{RDIST_{ij}} \right) \right) \quad (4.5)$$

For the current version of Visuset $SIZEX = 1280\,pixels$ and $SIZEY = 880\,pixels$, and the spring constant was set to 0.2. It should also be noted that the selected values for the ideal distance and spring constant K are related to the values chosen for $SIZEX$ and $SIZEY$ and the number of nodes and links in the system to be visualised. The stopping threshold can be set at any value, but from experimentation it was found that the number of nodes (as a pixel value) provided good operational results. Using Visuset it is also possible to disable the spring model so that the user can manually position nodes (and, if applicable, also change the size of individual islands at the same time). Further details concerning the background and development of Visuset can be found in [94].

In the proposed implementation of Visuset nodes are depicted as: (i) single nodes (i.e. self links where the "migration" is from and to the same node), (ii) node pairs linked by an edge, (iii) chains of nodes linked by a sequence of edges, or (iv) more complex sub-graphs (islands). The size (diameter) of the nodes indicates the number of elements represented by that node in M_{e_k} (the size of nodes at $M_{e_{k+1}}$ could equally well have been used, or some interpolation between M_{e_k} and $M_{e_{k+1}}$).

4.6.2 Animation of Pattern Migration

The animation mechanism, provided by Visuset, can be applied to pairs of visualisations (as described above) to illustrate the migration of patterns over three episodes (SOMs). Each visualisation is referred to as a mapping of the nodes in a SOM M_{e_i} to a SOM M_{e_j}. At the start of an animation the display will be identical to the first visualisation (Map1) and will move to a configuration similar to the second visualisation (Map 2), although nodes will not necessarily be in the same display location. Thus the animations show how subsequent mappings change and consequently how the pattern "communities" change. As the animation progresses the correlation coefficients (C-values) are linearly incremented or decremented from the values for the first map to that of the second map. Thus, as the animation progresses the links, nature of the islands, and overall number of nodes will change. For example if the correlation coefficient for a node in Map 1 is 0.3 and in Map 2 is 0.1 (assuming a threshold of 0.2) the node will "disappear" half way through the animation. Alternatively, if the correlation coefficient for a node in Map 1 is 0.1 and in Map 2 is 0.5 (again assuming a threshold of 0.2) the node will "appear" a quarter of the way through the animation. Nodes that disappear and appear are highlighted in white and pink respectively (nodes that persist are coloured yellow).

T2	T1 Node ID			Total
Node ID	1	2	3	
1	4	2	2	8
2	0	6	4	10
3	1	2	9	12
Total	5	10	15	30

Table 4.5: Pattern Migration Summary for Example Network Given in Figure 4.14

$T2$ Node ID	T Node ID	Patterns at $T1$ (P)	Patterns at $T2$ (Q)	Patterns Moved (X)	$P \times Q$	$\sqrt{P \times Q}$	$X \div \sqrt{P \times Q}$
1	1	5	8	4	40	6.32456	0.63246
1	2	5	10	0	50	7.07107	0.00000
1	3	5	12	1	60	7.74597	0.12910
2	1	10	8	2	80	8.94427	0.22361
2	2	10	10	6	100	10.00000	0.60000
2	3	10	12	2	120	10.95445	0.18257
3	1	15	8	2	120	10.95445	0.18257
3	2	15	10	4	150	12.24745	0.32660
3	3	15	12	9	180	13.41641	0.67082

Table 4.6: C-value (Correlation Coefficient) Calculation for Example Network Given in Figure 4.14

4.6.3 Worked Example of C-value Calculation

Figure 4.14 shows the migration of patterns through a three node network. The left hand network shows the state at time one (T_1) and the right hand network at time two (T_2). The nodes in each case are labeled with the number of patterns held at the node at these times. The middle network (in Figure 4.14) shows the number of patterns that have migrated to and from the nodes in the network from time T_1 to time T_2. Table 4.5 summarises this migration. The calculation of the C-values (correlation coefficients) for this network is given in Table 4.6. If a Min-Rel threshold of 0.2 is used (as advocated by experiments in this thesis), five of the migrations remain, as illustrated in Figure 4.15 (in the figure the arcs are labeled with the relevant C-values).

4.7 Discussions and Assumptions

The research work encapsulated by the modules used for FTPA are among the main contribution of the work described in this thesis. The proposed mechanisms provide support for the identification and analysis of trends in social network datasets. It is assumed that the datasets are arranged in episodes so that trend comparisons can be conducted. The discovered trend patterns are selected if at least one of their frequency counts is above the minimum threshold; the sequence of frequency counts is then used to define a “trend line”. As will be established by the evaluation described in the following chapter, using the identified trends users will be able to obtain useful knowledge so as to provide support for decision making.

The FPTA process addresses the research issues on pattern and trend representation, analysis and interpretation identified in Section 1.2. As will be demonstrated the proposed modules are able to handle large collections of episodes and interpret patterns and trends so as to highlight “interesting” and significant patterns and trends. The trend grouping and clustering methods allows users to view patterns and trends in a specific and more focused way. The process accentuates trend changes and pattern migration between episodes so as to support the identification of temporal changes in data. The Visuset software incorporated into the FPTA process presents networks of relationships between trend clusters and patterns so as to provide users with a clear illustration of these changes.

To maintain the flexibility and re-usability of the proposed modules, a number of assumptions were applied:

1. **Data format**: The datasets are in a binary valued format.
2. **Data granularity**: The time stamps and episodes are uncomplicatedly defined according to users’ needs and interests. (In this thesis the time stamps are assumed to represent months and the episodes years)
3. **Process sequence**: The input to each module (except the first module) is the output from previous module.

4.8 Summary

This chapter presented the FPTA process which comprises four of the modules included in the Predictive Trend Mining framework: (i) Trend Identification, (ii) Trend Grouping, (iii) Pattern Migration Clustering and (iv) Pattern Migration Visualisation. The objective of FPTA is to provide support for the identification of temporal patterns and trends, and provide support for their analyses. In the Trend Identication module, TM-TFP identifies frequent patterns or itemsets and determines the support values used to define trends. The Trend Grouping module groups similar types of trends and detects changes that may indicate “interesting” patterns and trends. The identified trend changes and pattern migrations are then used by the Pattern Migration Clustering and Visualisation modules. In the following chapter, the evaluation of the FTPA process is presented.

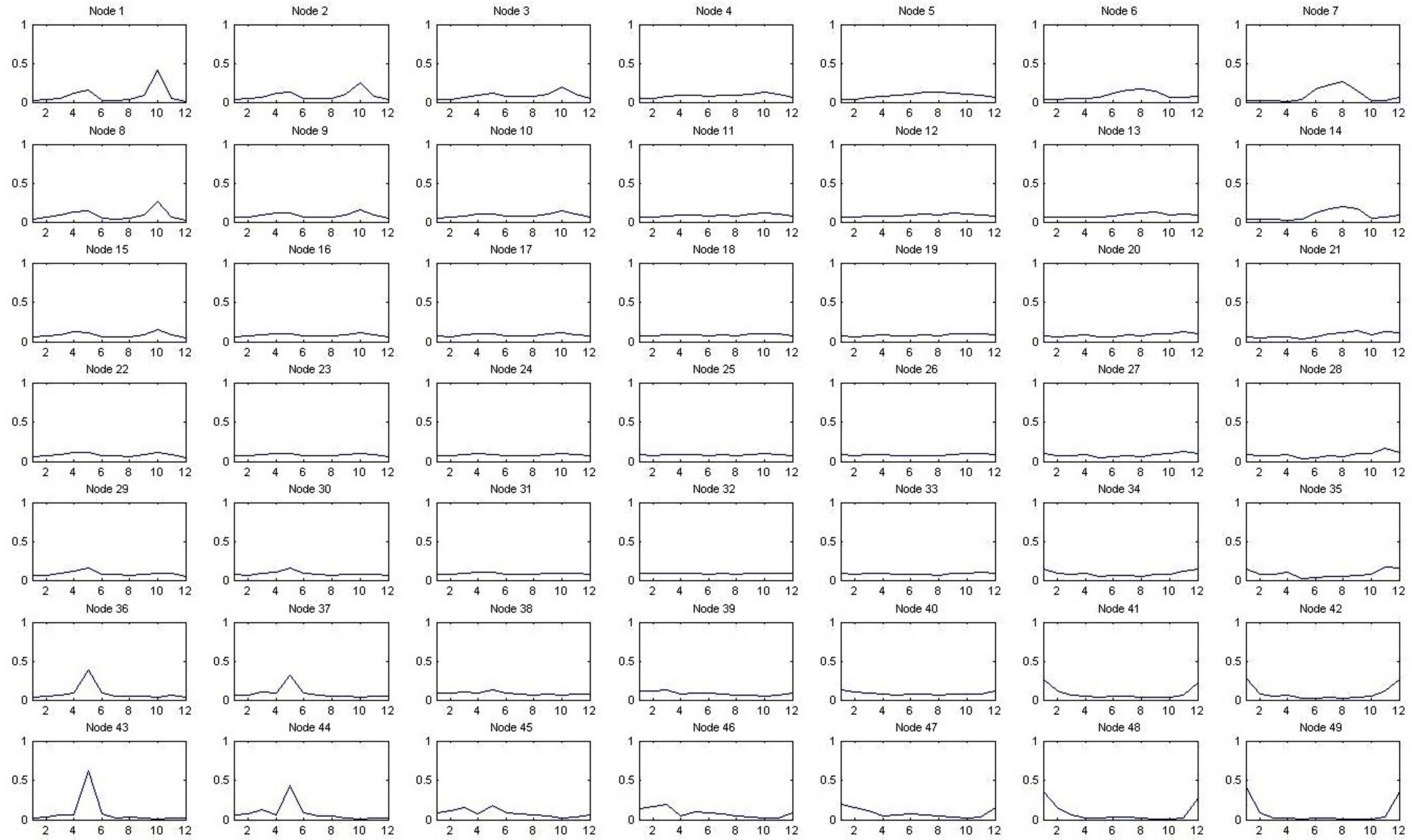


Figure 4.6: Prototype map with 7 × 7 map nodes

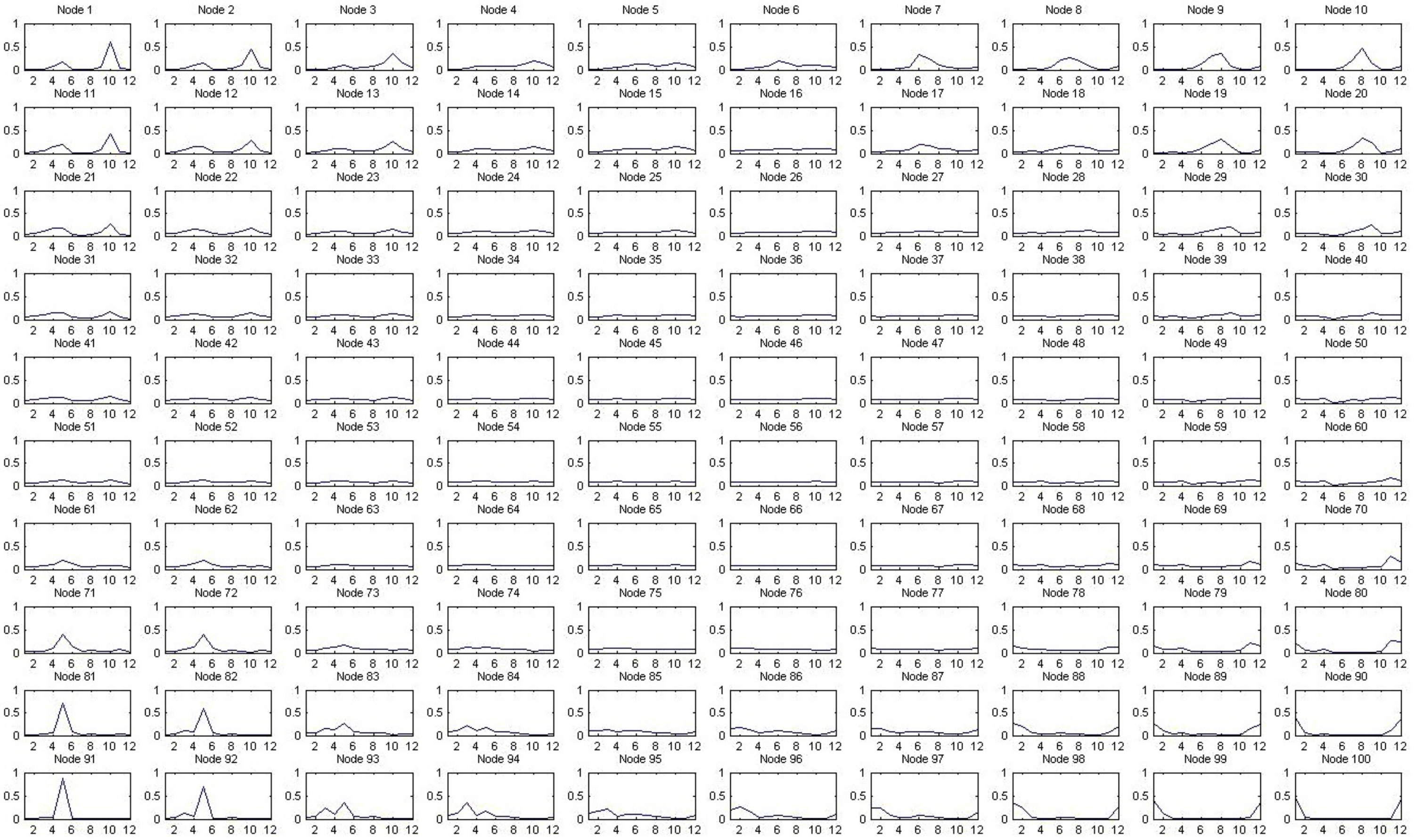


Figure 4.7: Prototype map with 10 × 10 map nodes

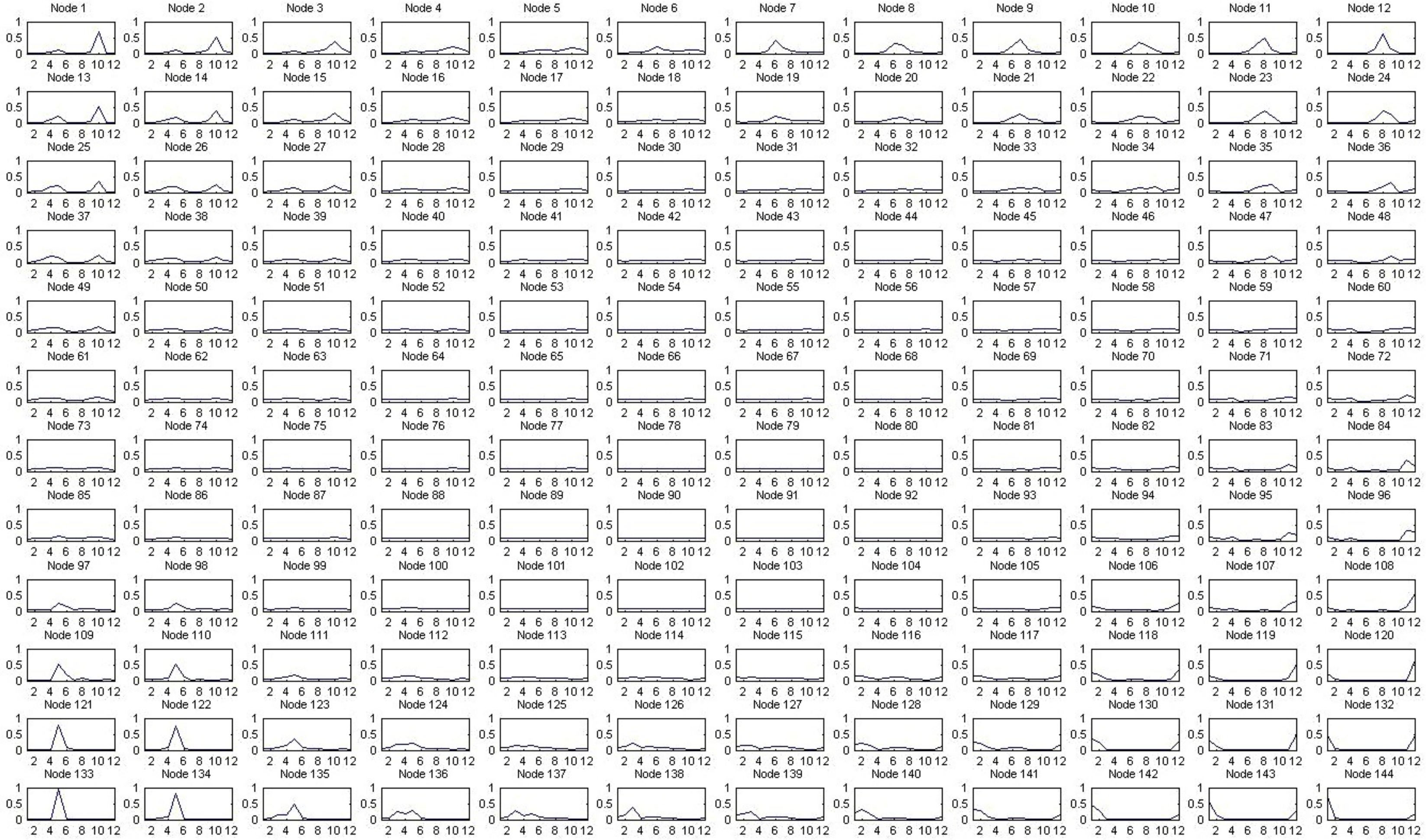


Figure 4.8: Prototype map with 12 × 12 map nodes

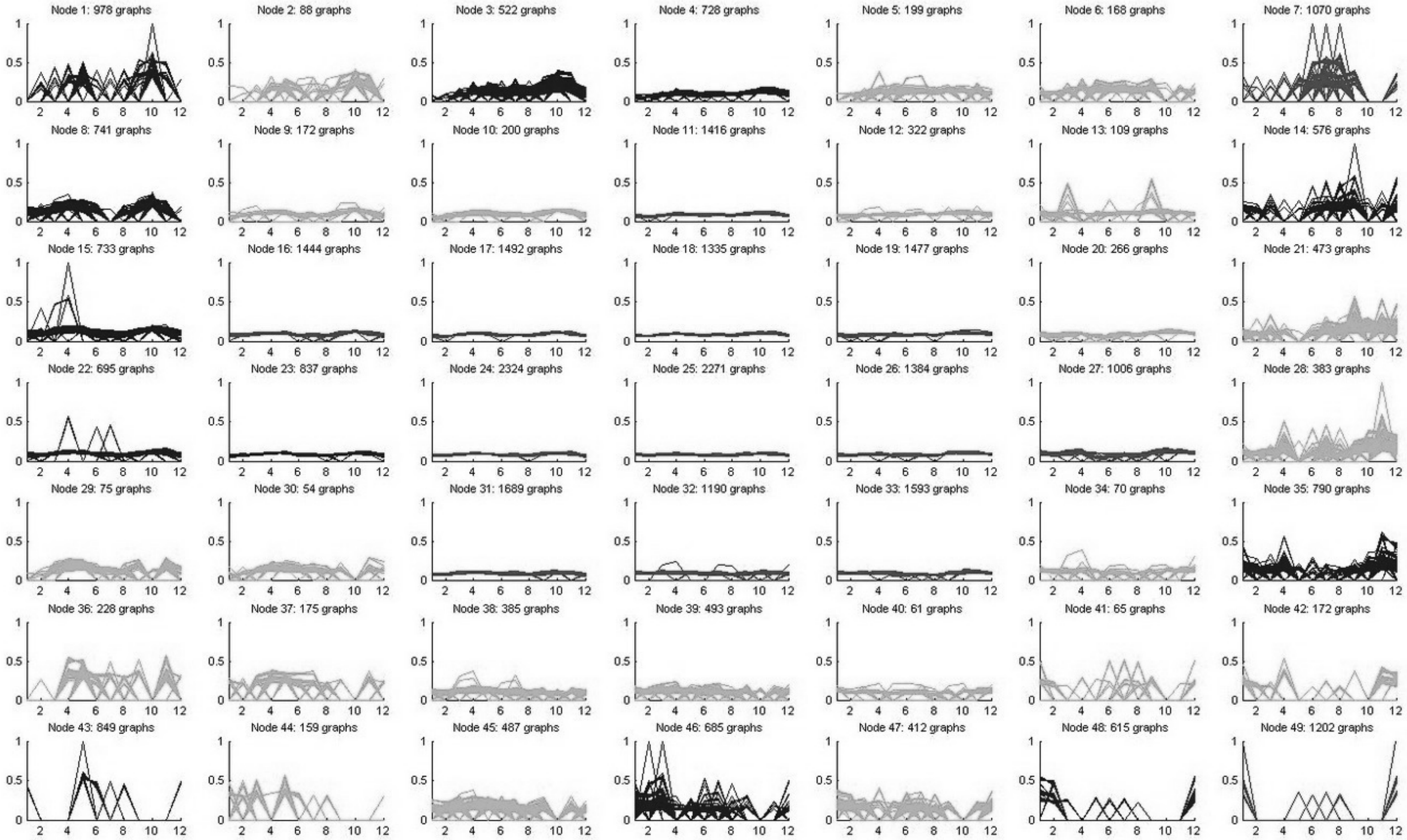


Figure 4.9: Trend line map with 7 × 7 map nodes

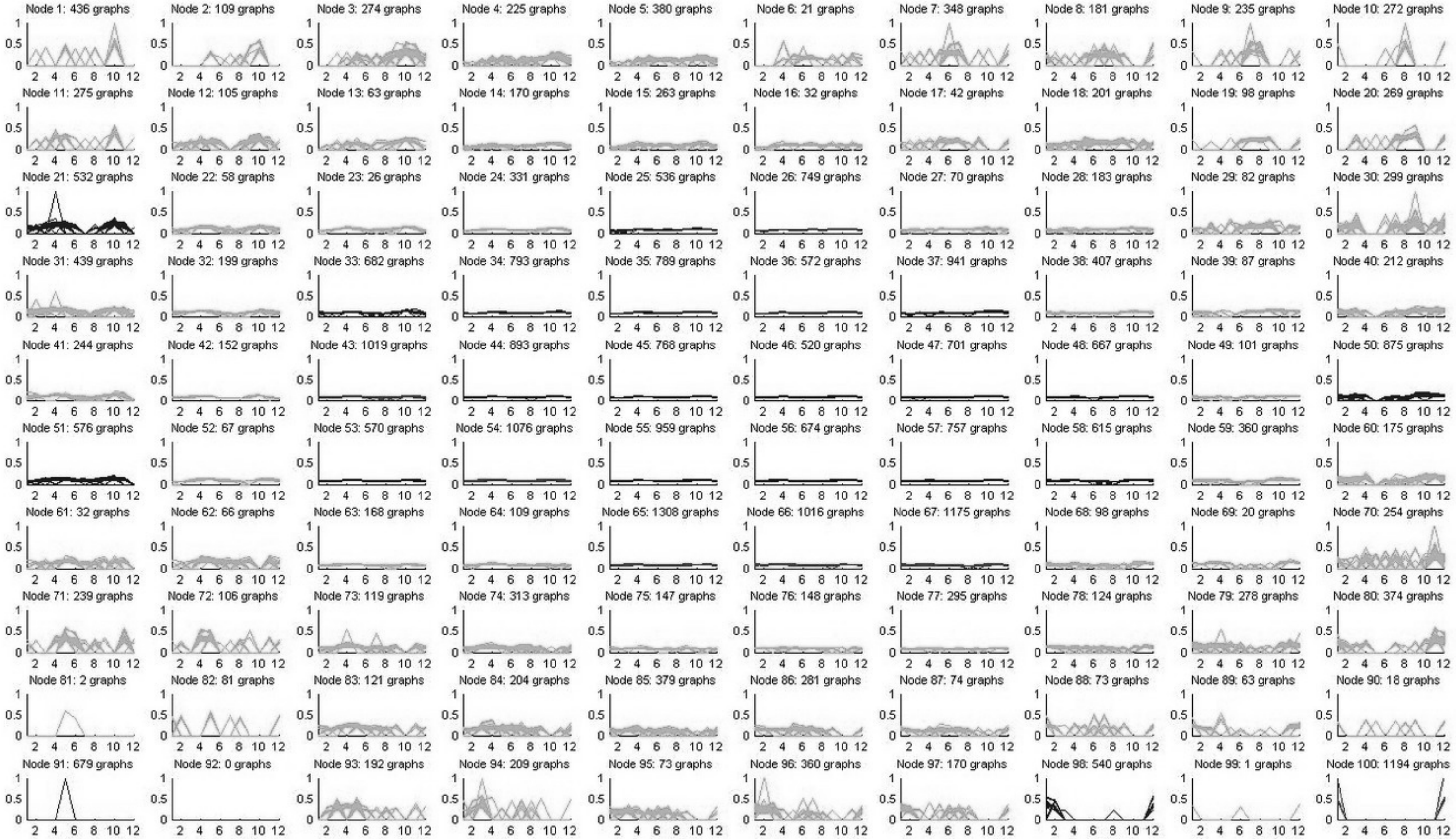


Figure 4.10: Trend line map with 10 × 10 map nodes

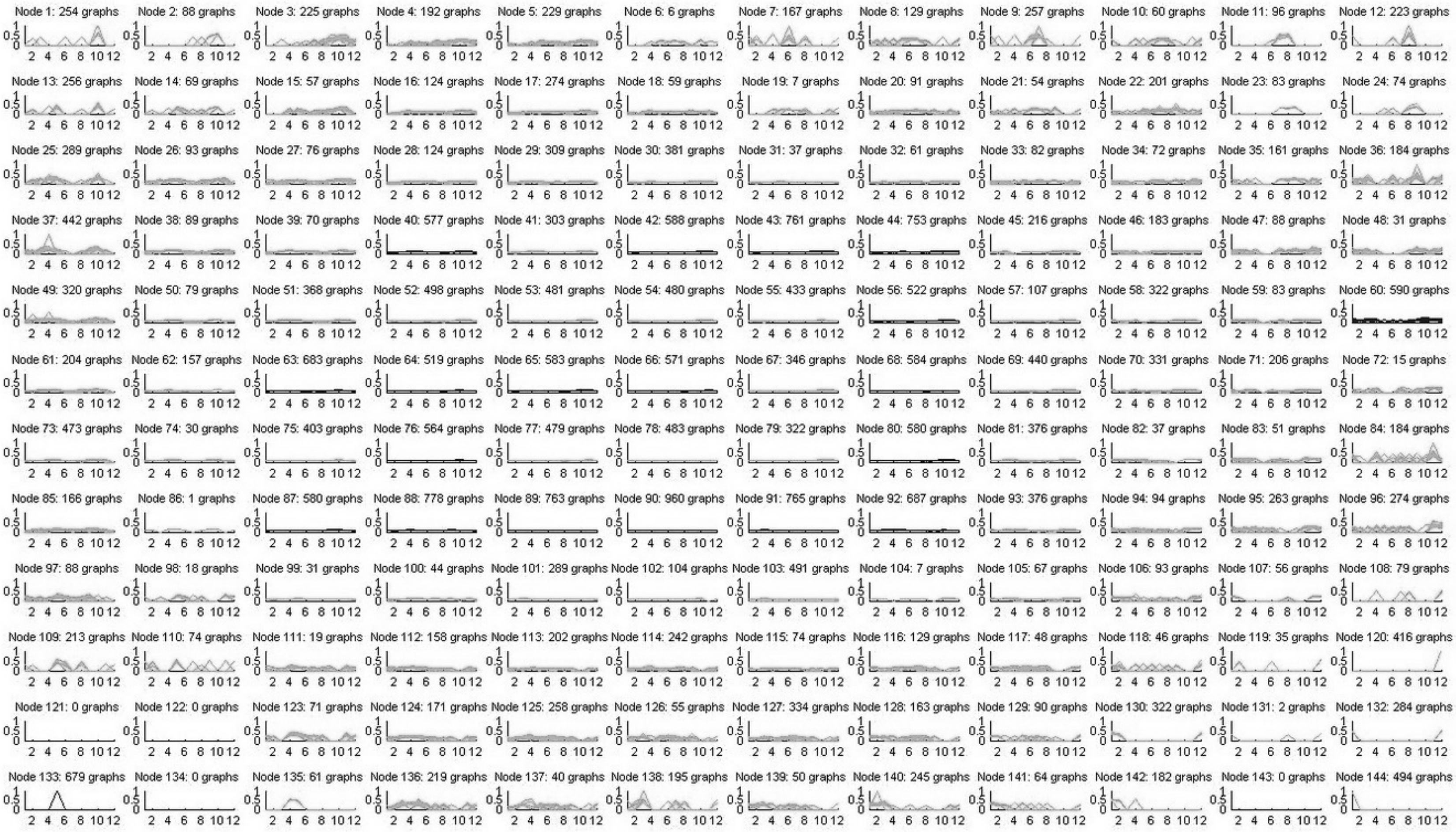


Figure 4.11: Trend line map with 12 × 12 map nodes

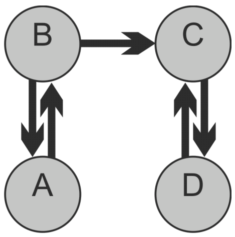


Figure 4.12: Four Node Example Network

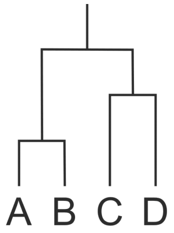


Figure 4.13: Dendrogram for Hierarchical Clustering Example

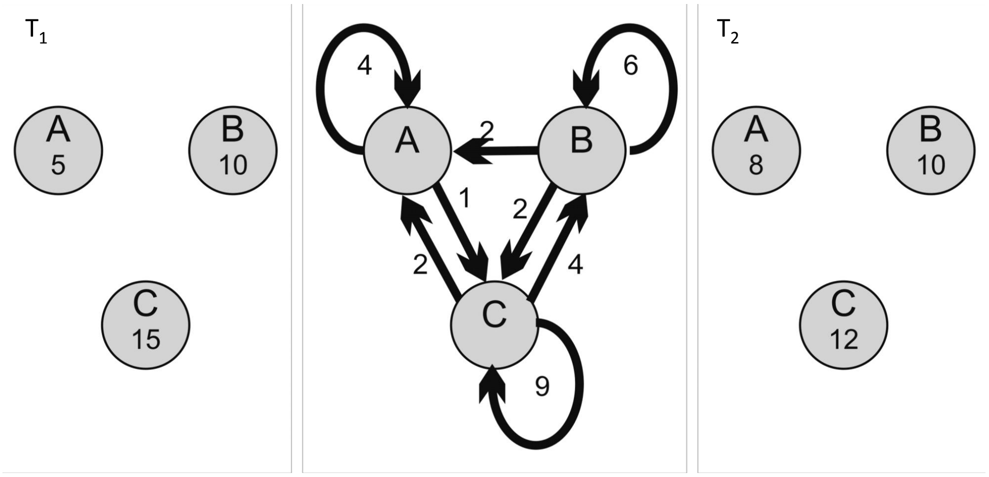


Figure 4.14: Three Node Example Network Showing Pattern Migrations from T_1 to T_2

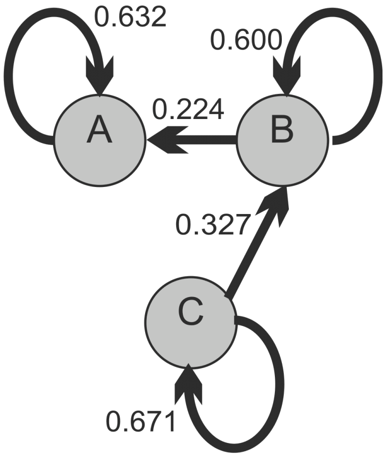


Figure 4.15: Three Node Example Network with Irrelevant links Removed

Chapter 5

Evaluation of The Frequent Pattern Trend Analysis Mechanism

In this chapter the author presents the results of a number of experiments conducted to evaluate the Frequent Pattern Trend Analysis element of the research described in Chapter 4. From Chapter 4 it will be recalled that the proposed mechanisms encompasses: (i) Trend Identification, (ii) Trend Grouping, (iii) Pattern Migration Clustering and (iv) Pattern Migration Visualization. The experiments described in this chapter were designed to demonstrate that the proposed mechanisms served to achieve the research objectives at which they were directed, and contributed to the provision of an answer to the overall research question as presented in Chapter 1. More specifically the intention is to demonstrate the flexibility, reusability and effectiveness of the Trend Identification, Trend Grouping, Pattern Migration Clustering and Pattern Migration Visualisation modules.

The evaluation of each module is described in sequence according to the order in which each is applied in the Predictive Trend Mining Framework (PTMF). The evaluation was conducted using the three social network datasets presented in Chapter 3. Each dataset was preprocessed in the same manner.

All three datasets provided consistent results to support the research objectives. A large number of frequent patterns and trends were identified for all the network datasets using several minimum support thresholds. In each case these trends were then grouped into trend clusters using a SOM. The resulting SOM maps were then analysed so as to identify pattern migrations. Lastly the visualisation module was applied to illustrate the captured pattern migration information in the form of network maps for display to end users.

The remainder of this chapter is organized as follows. In Section 5.1 the experimental analysis of the Trend Identification module using the three selected social network datasets is discussed. Section 5.2 presents a demonstration of the Trend Grouping module when applied to large numbers of discovered frequent patterns and trends. Section 5.3 then discusses and analyses the pattern migration and the identification of trend clusters that involve pattern migrations. In Section 5.4, a discussion is presented that considers the experimental evaluation of the Pattern Migration Visualisation module. Finally, in Section 5.5, the chapter is concluded with a brief summary.

5.1 Experimental Analysis of The Trend Identification Module

In general, the modules for Frequent Pattern Trend Analysis (FPTA) can be applied to social network data of all kinds. However, as noted in earlier chapters, the work in this research focuses on business community social networks. This section describes the evaluation of the Trend Identification module using the CTS social network, Deeside Insurance social network and MAF Logistic Cargo social network introduced earlier. The Trend Identification module was designed to take input in a standard binary valued format. The idea being that this would allow for general applicability (as confirmed by the use of the three different datasets for the evaluation described here). Each of the experiments assumed twelve time stamps per data episode (e) where each time stamp represented a month of data. More specifically: (i) the CTS network has 4 × 12 time stamps, (ii) the Deeside Insurance network has 2 × 12 time stamps, and (iii) the MAF Logistic Cargo network has 2 × 12 time stamps.

The TM-TFP algorithm operates using a minimum support thresholds, α, to identify the frequent patterns. As mentioned earlier, low support threshold values are desirable so as to make sure that no "interesting" patterns are missed. Thus a range of α values were considered with respect to each dataset. In the following sub-sections, Sub-section 5.1.1 provides the Trend Identification results from the CTS network, then Sub-section 5.1.2 discusses the trend identification results using the Deeside Insurance network followed by the results obtained using the MAF Logistic Cargo network in Sub-section 5.1.3. In each case the analysis was conducted in terms of the number of identified frequent patterns and the run time. Some experimental analysis using attribute feature constraints are then presented in Sub-section 5.1.4. The section is concluded with a brief summary (Sub-section 5.1.5).

5.1.1 GB Cattle Movement Trend Identification

For experimental purposes, using CTS dataset, three support threshold values of 0.5%, 0.8% and 1.0% were used. The number of identified frequent pattern trends in each case is presented in Table 5.1 (a similar table was presented in Table 4.1 in Chapter 4). From the tables it can be seen that large numbers of trends are discovered. For example, using a support threshold of 0.5%, the number of identified trends discovered

Episode	Support Threshold		
(year)	0.5%	0.8%	1.00%
2003	63,117	34,858	25,738
2004	66,870	36,489	27,055
2005	65,154	35,626	25,954
2006	62,713	33,795	24,740
Average	64,464	35,192	25,872

Table 5.1: Number of trends identified using TM-TFP for a sequence of four CTS network episodes and a range of support thresholds

Episode	Support Threshold		
(year)	0.5%	0.8%	1.00%
2003	97.02	69.49	63.54
2004	92.44	70.0	64.0
2005	83.25	59.55	53.5
2006	101.06	72.95	66.09
Average	93.44	68.00	61.78

Table 5.2: The TM-TFP algorithm run time values (seconds) using the CTS social network episodes

over the four episodes (2003, 2004, 2005 and 2006) were 63117, 66870, 65154 and 62713 respectively. The number of frequent patterns discovered, as expected, increases as the support threshold decreases. The lower the support threshold the greater the number of discovered frequent patterns and hence the greater the number of trends. Thus the use of a low support threshold ensures that no potentially interesting trends are omitted. The number of frequent patterns and trends are significantly reduced when a minimum support of 1.0% is used. Nevertheless, the identified "super" set of frequent patterns discovered when using 0.5%, 0.8% and 1.0% are similar. For completeness, Table 5.2 shows the run time values for identifying frequent patterns and trends so as to give an indication of the time complexity of the TM-TFP algorithm. From the table it can be seen that increases in the minimum support thresholds results in corresponding linear decreases in the TM-TFP run time.

Some examples of the sort of the frequent patterns that may be extracted from the CTS network are presented in Table 5.3. Simmental cattle are a versatile breed of cattle from Switzerland often crossed with other breeds. The patterns include: information on animal age, animal gender, breed name, breed type (dairy or beef), number of cattle moved, and sender and receiver location type and grid square area. It should be noted that the first two frequent patterns in Table 5.3 include "zero" support values in the trend definition. It should be recalled that this is not a real zero, but indicates that the

No.	Frequent Patterns	Trends
1.	$\{2year\ old \leq AnimalAge \leq 5year\ old, Breed = Friesian,\ Breed\ Type = dairy, Receiver\ Location\ Type = Slaughter\ House(RedMeat)\}$	$\{2765, 2211, 2562, 3279, 0, 1307, 2004, 1906, 2593, 3315, 3391, 3152\}$
2.	$\{Gender = female, 2year\ old \leq Animal\ Age \leq 5year\ old, Breed = Friesian, Breed\ Type = dairy, ReceiverLocationType = Slaughter\ House(RedMeat)\}$	$\{2741, 2193, 2541, 3251, 0, 1295, 1995, 1896, 2581, 3299, 3384, 3145\}$
3.	$\{Gender = female, Breed = Simmental\ Cross, Breed\ Type = beef\ and\ dairy, Receiver\ Location\ Type = Slaughter\ House\ (RedMeat)\}$	$\{4050, 3322, 3175, 3690, 2777, 2722, 2972, 2494, 3082, 3823, 3951, 3717\}$
4.	$\{Breed\ Type = beef, SenderArea = 13, Receiver\ Location\ Type = Slaughter\ House\ (RedMeat)\}$	$\{1786, 1593, 1553, 1736, 1410, 1291, 1541, 1369, 1839, 2000, 1772, 1694\}$
5.	$\{AnimalAge \leq 1yearold, Breed\ Type = beef, ReceiverArea = 14, ReceiverLocationType = Agricultural\ Holding, Number\ Cattle\ Moved \leq 5\}$	$\{2098, 1925, 2854, 3051, 3364, 2705, 2793, 2469, 3018, 3189, 3031, 2336\}$

Table 5.3: Example frequent patterns and associated trends obtained from the 2003 CTS network using a 0.5% minimum support threshold

support value at the associated time stamp dropped below α (see discussion in Section 4.3).

Figure 5.1 presents a visualisation of the trends given in Table 5.3. The trend lines in the figure show that it is possible to identify a variety of types of trend line in the CTS data. The trend lines illustrate that the monthly frequency occurrences for each pattern fluctuate throughout the year. In certain months the trends experienced sharp rises and dips below the α threshold (trend lines 1 and 2). Note that the reason why trends 1 and 2 are similar is that almost all dairy animals are between two and five years of age and are female. Male animals (if not eaten earlier on) reach a much greater age.

5.1.2 Deeside Insurance Quotation Trend Identification

The Deeside Insurance social network was used to demonstrate the general applicability of the TM-TFP algorithm with respect to alternative types of social network data to that described by the CTS dataset (recall that the Deeside Insurance dataset comprises a star network while the CTS network is a described as a complex star network). Table 5.4 presents the number of trends generated by applying TM-TFP to the Deeside Insurance dataset using a range of support thresholds of 2%, 3% and 5% respectively. Higher support thresholds were used than in the case of the CTS dataset because the

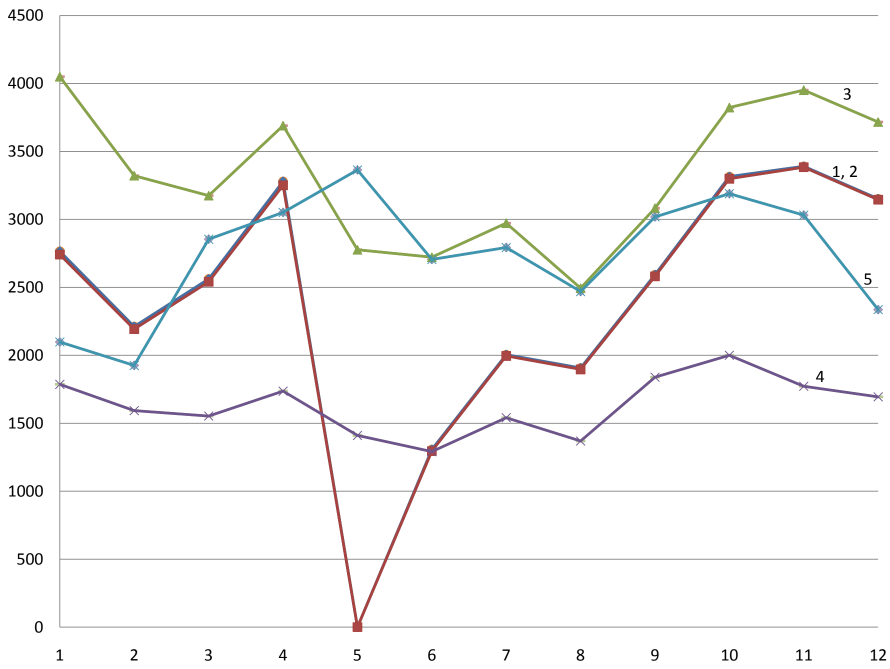


Figure 5.1: Trend lines for the CTS Frequent Patterns given in Table 5.3

Deeside Insurance dataset was smaller (in terms of number of records).

Although the Deeside Insurance dataset was smaller than the CTS dataset, and higher values for α were used, a larger number of patterns and consequently trends were still identified using the TM-TFP algorithm. The reason behind this is that the Deeside Insurance social network has more data attributes compared to CTS social network. Thus the number of attributes in the input data is an important contributing factor with respect to the number of trends identified. Table 5.5 presents the recorded run time values obtained when identifying patterns and trends using the Deeside Insurance dataset.

Episode	Support Threshold		
(year)	2%	3%	5%
2008	314471	142175	55241
2009	284871	122371	49983
Average	299671	132273	52612

Table 5.4: Number of frequent pattern trends identified using the Deeside Insurance network and a range of support thresholds

Table 5.6 shows some examples of frequent patterns and trends identified using the Deeside Insurance dataset. Again, the trends associated with each pattern comprise 12 frequency counts. The frequent patterns attributes include: the length of disquali-

Episode	Support Threshold		
(year)	2%	3%	5%
2008	23.43	12.8	5.67
2009	23.42	11.77	4.99
Average	23.43	12.29	5.33

Table 5.5: The TM-TFP algorithm run time values (seconds) using the Deeside Insurance social network

No.	Frequent Patterns	Trend
1.	$\{Length\ of\ disqualification \leq 5,$ $0 \leq Convict\ code\ number \leq 50,$ $Convict\ code = NULL, 0 \leq Postcode\ sector \leq 10,$ $2001 \leq Year\ of\ manufactured \leq 2006\}$	{40, 51, 49, 49, 65, 54, 64, 72, 90, 102, 80, 61}
2.	$\{Fault = yes, Length\ of\ disqualification \leq 5,$ $Convict\ code = NULL, 0 \leq Postcode\ sector \leq 10,$ $0 \leq Engine\ size \leq 1000\}$	{28, 18, 21, 35, 27, 28, 34, 54, 82, 51, 54, 30}
3.	$\{Fault = yes, Length\ of\ disqualification \leq 5,$ $Convict\ code = NULL, 0 \leq Postcode\ sector \leq 1,$ $1601 \leq Engine\ size \leq 2000\}$	{21, 20, 31, 32, 28, 25, 28, 42, 70, 42, 40, 32}
4.	$\{0 \leq Penalty \leq 1000, Length\ of\ disqualification \leq 5,$ $Convict\ code = SP, 0 \leq Postcode\ district \leq 10,$ $Postcode\ area = CH, Engine\ size \leq 2001\}$	{20, 22, 24, 45, 19, 33, 29, 37, 37, 0, 23, 25}
5.	$\{Penalty \leq 2001, Length\ of$ $disqualification \leq 5,$ $0 \leq Postcode\ sector \leq 10\}$	{13, 0, 14, 0, 0, 19, 25, 0, 0, 0, 23, 24}

Table 5.6: Example frequent patterns and associated trends obtained from the 2008 Deeside Insurance network using a 5% minimum support threshold

fication, conviction code and number, car year of manufacture and customer postcode. Customer postcode is a spatial attribute that indicates the geographic distribution of the Deeside Insurance network. Again in the last two examples shown in Table 5.6, the trend lines sometimes drop below α.

Figure 5.2 shows the trend lines associated with the examples of frequent patterns presented in Table 5.6. Three of the trend lines shown in the figure have a steady increase and peak in September and October and dip sharply in November and December; thus indicating a high demand for insurance quotes in September and October. Trends 4 and 5 have support values that fall below α as indicated by the zero values for certain months.

5.1.3 MAF Logistic Cargo Distribution Trend Identification

The experiments conducted using the MAF Logistic Cargo network were principally designed to corroborate the results produced using the CTS and Deeside Insurance

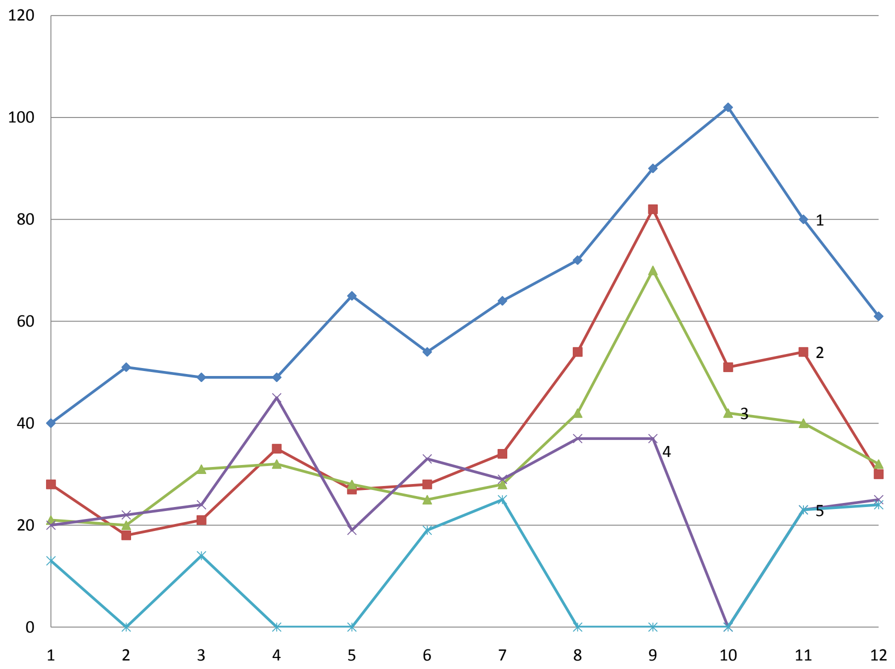


Figure 5.2: Trend lines for the Deeside Insurance Frequent Patterns given in Table 5.6

networks. The major difference between the MAF Logistic Cargo dataset and the other two datasets, however, is that the number of records is far fewer compared to the other two cases. Nevertheless, the results obtained demonstrated that even with a small number of data records, a large number of frequent patterns and trends can still be identified if the dataset features a significant number of attributes.

Episode	Support Threshold		
(year)	2%	3%	5%
2008	3491	3491	3491
2009	2761	2761	2609
Average	3126	3126	3050

Table 5.7: Number of frequent pattern trends identified using the MAF Logistic Cargo network and a range of support thresholds

Three minimum support threshold values of 2%, 3%, and 5% were used. Table 5.7 shows the number of frequent patterns and trends identified using the MAF Logistic Cargo network dataset. Because of the small number of records (average of approximately 40 per time stamp) the same number of patterns are discovered in both the 2008 and the 2009 datasets regardless of the minimum support threshold (α value) used. Table 5.8 presents the recorded run times (in seconds) obtained when applying TM-TFP to identify the patterns and trends in the MAF Logistic Cargo dataset.

Episode	Support Threshold		
(year)	2%	3%	5%
2008	0.27	0.25	0.23
2009	0.27	0.26	0.24
Average	0.27	0.26	0.24

Table 5.8: The TM-TFP algorithm run time values (seconds) using the MAF Logistic Cargo network network

No.	Frequent Patterns	Trend
1.	$\{Logistic\ items = Ordnance\ items\}$	{3, 7, 3, 2, 6, 1, 3, 1, 3, 3, 2, 1}
2.	$\{Sender\ city = Batu\ Caves,$ $Logistic\ items = 1\ tonne\ truck\}$	{0, 2, 3, 1, 1, 4, 1, 0, 0, 0, 0, 0}
3.	$\{Sender\ city = Batu\ Caves,$ $Sender = 92\ DKP\}$	{0, 11, 8, 4, 1, 8, 3, 0, 3, 1, 3, 0}
4.	$\{Sender\ city = Batu\ Kentonmen, Sender = 91\ DPO,$ $Logistic\ items = Ordnance\ items\}$	{3, 0, 3, 2, 2, 1, 3, 1, 3, 3, 1, 1}
5.	$\{Receiver = 5\ KOD, Sender = 91\ DPO\}$	{1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1}
6.	$\{Receiver = 5\ KOD, Sender\ city = Batu\ Kentonmen,$ $Sender = 91\ DPO, Logistic\ items = Ordnance\ items\}$	{1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1}
7.	$\{Receiver\ city = Sibu, Receiver = 9\ KOD,$ $Sender\ city = Batu\ Kentonmen\}$	{2, 0, 0, 2, 1, 0, 1, 0, 0, 2, 0, 2}
8.	$\{MYR50001 \leq Shipment\ cost \leq MYR100000,$ $Receiver\ city = Sibu, Sender\ city = Batu\ Kentonmen,$ $Sender = 91\ DPO, Logistic\ items = Ordnance\ items\}$	{0, 0, 2, 1, 0, 0, 1, 0, 0, 0, 1, 0}

Table 5.9: Example frequent patterns and associated trends obtained from the 2008 MAF Logistic Cargo network using a 5% minimum support threshold

Table 5.9 provides some examples of frequent patterns and trends extracted from the MAF Logistic Cargo network. The frequent patterns feature the following attributes: logistic item, shipment cost, sender ID, receiver ID and city location of sender and receiver. Again, the identified trends in the MAF Logistic Cargo frequent patterns have several support values below α. Figure 5.3 illustrates the trend lines associated with the frequent pattern examples given in Table 5.9. From the figure it can be seen that the trend lines fluctuate with sharp increases and drops; again it is difficult to interpret and analyse the trends. Thus it was deemed desirable to have a clustering method for grouping the discovered trend lines so as to provide support for further analysis.

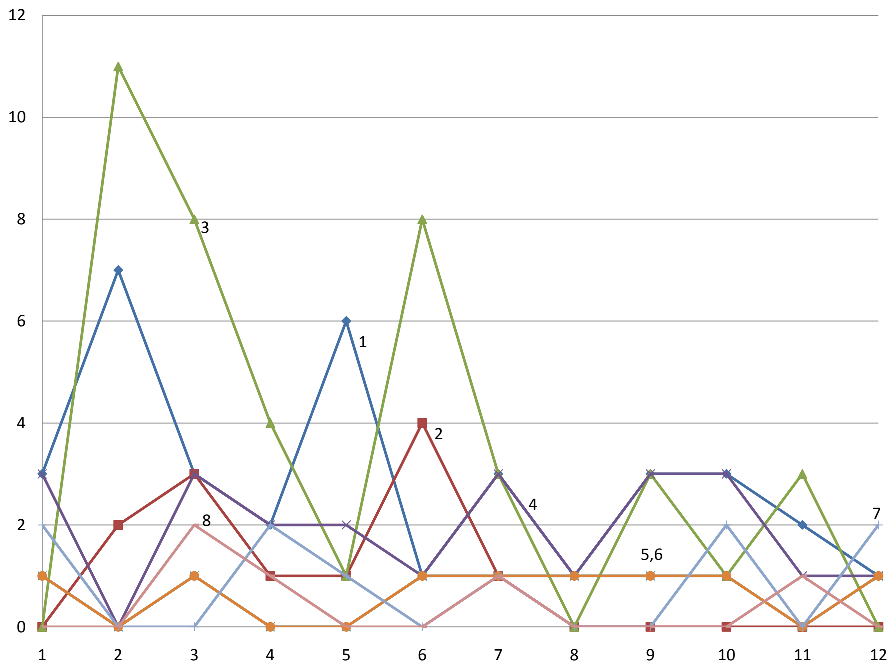


Figure 5.3: Trend lines for the MAF Logistic Cargo Frequent Patterns given in Table 5.6

5.1.4 Experimental Analysis of Trend Identification with Constraints

From the above a large number of trends are typically discovered. This hampers their analysis. One way of supporting the desired analysis is the proposed clustering facility to group similar trends. Another way of reducing the overall number of trends is to apply some form of constraints to the input data. To determine the effect of using constraints a number of experiments were conducted using the CTS and Deeside Insurance networks datasets. The constraints are subjective according to nature of the users' interests. However by using constraints, it is possible to reduce the number of discovered patterns and trends and thus reduces the overall complexity of the findings. Note that the analysis of the use of constraints presented here has been previously published by the author in [98].

For the analysis using the CTS network, in the context of constraints, the author applied several pattern constraints. In the reported experiments, three pattern constraints were applied:

$$\text{Constraint 1: } \{Breed\ Type = Beef\}$$
$$\text{Constraint 2: } \{Breed\ Type = Dairy\}$$
$$\text{Constraint 3: } \{Sender\ Location\ Type = $$
$$Agricultural\ holdings,\ Receiver\ Location\ Type = Market\}$$

The effect of Constraint 1 and Constraint 2 is that only records with cattle used for

Beef or *Dairy* respectively are considered. Whereas, Constraint 3 is designed to select records describing cattle movements from *Agricultural holdings* to *Markets*. Table 5.10 presents the number of patterns and trends discovered using all the available data and with the above constraints applied. As in the previously reported experiments, as the support threshold increases, the number of identified patterns and trends decreases which may ease the process of interpretation of patterns and trends. However, it can be noted that the use of constraints serves to reduce to overall number of patterns (trends) to be considered when conducting further analysis.

Support Threshold (%)	No Constraint	Constraint 1	Constraint 2	Constraint 3
1	25736	2333	2583	1195
2	8945	1019	1181	535
3	4393	541	715	383
4	2739	349	483	311
5	1928	249	339	267

Table 5.10: Number of identified patterns using CTS network

Similarly, for the analysis using the Deeside Insurance network, the author also applied a number of pattern constraints so as to reduce the number of patterns to be considered to a more manageable number. The following three pattern constraints were applied:

$$\text{Constraint 1: } \{DriverAge = \{24 : 40\}\}$$
$$\text{Constraint 2: } \{Gender = Male\}$$
$$\text{Constraint 3: } \{PostcodeArea = CH\}$$

Constraint 1 has the effect of insisting that frequent patterns include the attribute $DriverAge = \{24 : 40\}$ (age between 24 and 40), while Constraint 2 has the effect of limiting the set of frequent patterns to those where *Gender* has the value *Male*. Constraint 3 has the effect of restricting patterns to those that include the *PostcodeArea* "CH" (Chester).

Support Threshold (%)	No Constraint	Constraint 1	Constraint 2	Constraint 3
1	830306	8239	5621	3965
2	206219	2163	1431	1595
3	94369	1038	677	863
4	55445	669	401	563
5	37239	469	283	427

Table 5.11: Number of identified trends using Deeside Insurance network

Table 5.11 gives the number of patterns and trends discovered using all the Deeside Insurance data; and with the application of each of the constraints. As expected, with low support thresholds, a large number of trends are generated in each case. When a constraint is imposed, the number of records to be considered is substantially reduced therefore fewer trends are discovered.

5.1.5 Trend Identification Summary

From the foregoing reported experiments conducted regarding the Trend Identification module, it can be deduced that, regardless of the sizes of the network the module is able to identify large numbers of frequent patterns trends (assuming a suitable α value is used). Because of the large numbers of patterns and trends that are discovered using TM-TFP, it is suggested that it will be difficult to analyse further the significance of the discovered patterns and trends without conducted some additional processing on the identified trends first. To aid the desired interpretation the use of constraints was suggested. To further aid interpretation trend grouping was proposed. An analysis of the Trend Grouping module is presented in the next section.

5.2 Experimental Analysis of The Trend Grouping Module

This section reports on the experimental analysis of the Trend Grouping module using the patterns and trends discovered with three different α values: (i) 0.5% for CTS, (ii) 5% for Deeside Insurance and (iii) 5% for MAF Logistic Cargo. The objective of the Trend Grouping module is to cluster similar identified trends so as to facilitate their analysis. SOM technology was proposed to identify the desired trend clusters. The SOM grid parameters (7×7 *and* 10×10) were defined according to the number of the discovered patterns and trends. Experiments have been done to determine the SOM grid parameters as described in Chapter 4. Recall that each node in the SOM map describes a trend cluster. Details of the frequent patterns (pattern codes and the count of the patterns) associated with each trend cluster are maintained.

Figure 5.4 provides the run time figures for grouping all three networks' trends (details of numbers of trends and α were given in Tables 5.1, 5.4 and 5.7). The chart in figure 5.4 shows that Deeside's trends took the longest run time to be grouped as a larger number of patterns were discovered using $\alpha = 2\%$ and 3%.

In the following sub-sections, the results of experiments to group the identified trends are presented with respect to the CTS in Sub-section 5.2.1, Deeside Insurance in Sub-section 5.2.2 and MAF Logistic Cargo in Sub-section 5.2.3. Some findings using the concept of constraints, which has previously been report in [98], are also presented in Sub-section 5.2.4. A brief summary of this section is presented in Sub-section 5.2.5.

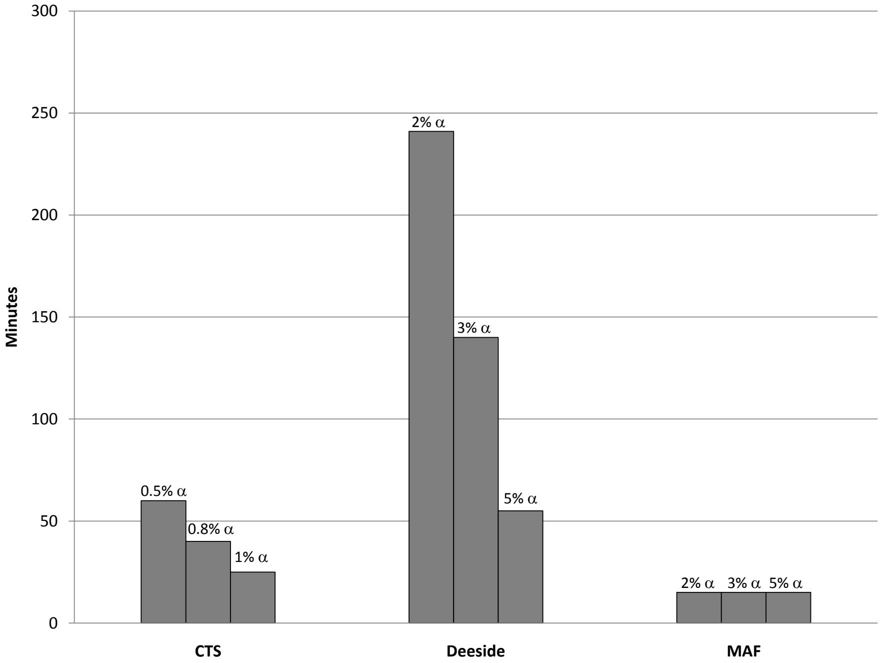


Figure 5.4: Trend grouping module run time (minutes) for the CTS, Deeside Insurance and MAF Logistic Cargo networks

5.2.1 GB Cattle Movement Trend Grouping

With respect to the CTS dataset, to identify the groupings within the collection of trends identified using TM-TFP, the SOM software was initialising with a 10×10 node map, and trained using the frequent pattern trends produced with the (earliest) 2003 episode. The resulting prototype map is shown in Figure 5.5. The prototype map groups similar trends occurring in the 2003 episode so that seasonal movement variations may be identified. For example: node 34 describes trends where the associated pattern is more prevalent in March, June and October; nodes 44 and 54 both describe trends where the associated pattern occurs frequently in spring and autumn; and so on. Analysis of the prototype map indicates, as might be expected, that hierarchies of patterns, comprising collections of sub-sets of a "parent" pattern, tend to appear in the same clusters. Recall also that the proximity between SOM nodes indicates the similarity between them; the greatest dissimilarity is thus between nodes at opposite ends of the diagonals in the SOM map.

Once the initial prototype map had been generated a sequence of trend line maps was produced, one for each episode. Figure 5.6 gives the map for the 2003 trend lines and Figure 5.7 that for the 2004 trend lines. Note that in Figures 5.6 and 5.7 each node has been annotated with the number of trends in the "cluster", and that nodes with "darker" trend lines indicate a greater number of lines within that cluster than nodes with "lighter" trend lines.

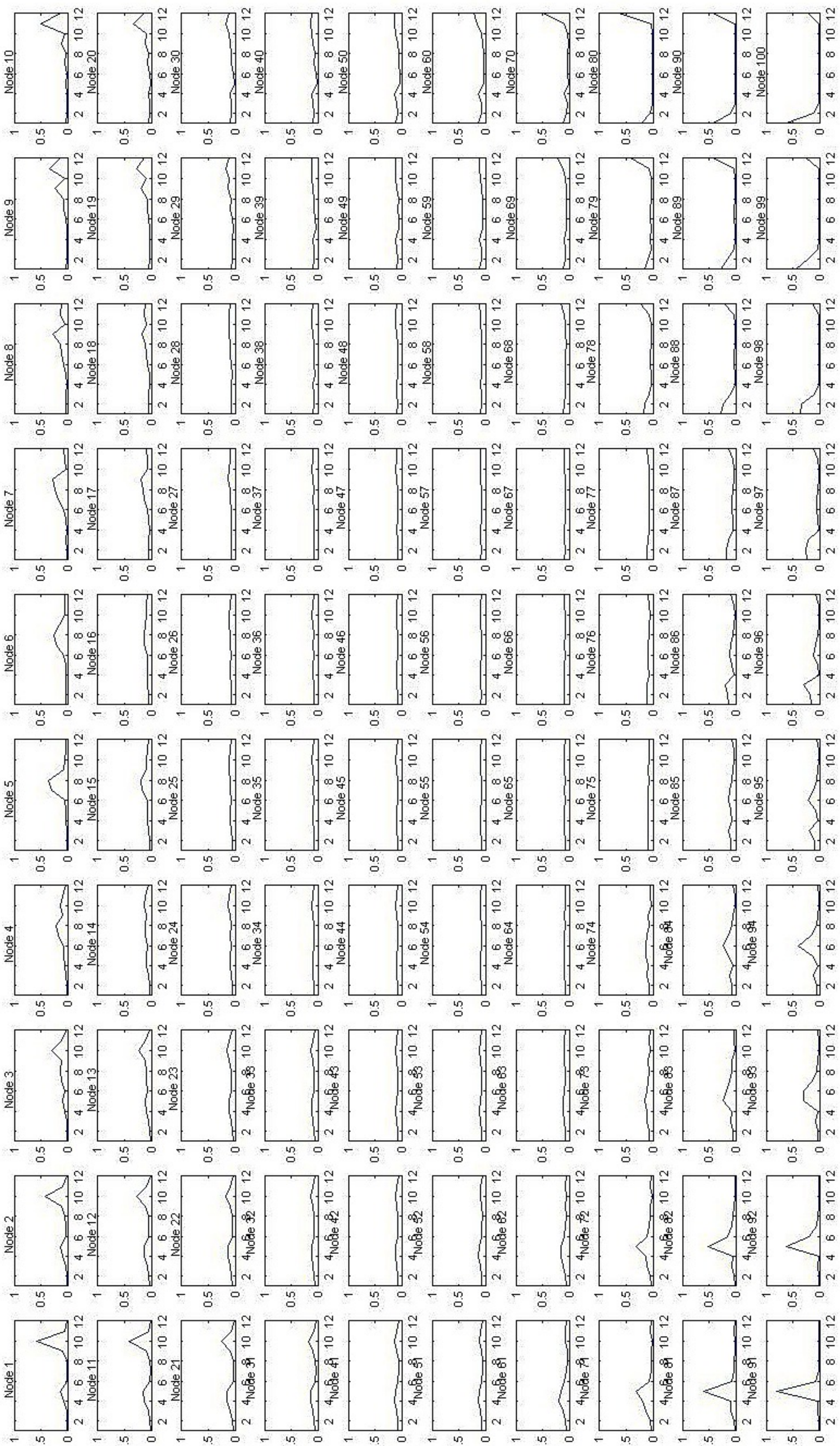


Figure 5.5: CTS network prototype map generated using 2003 episode

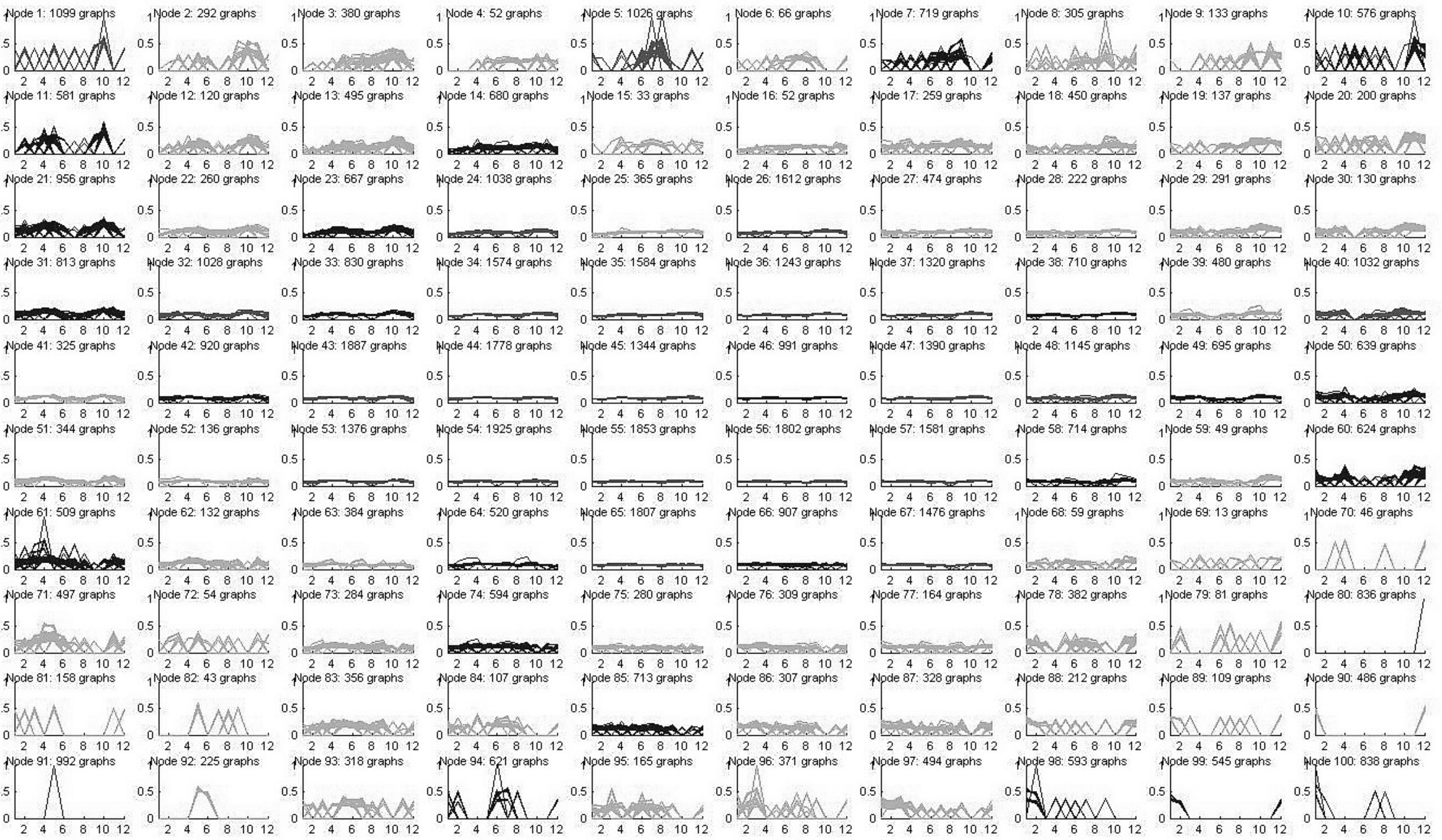


Figure 5.6: CTS network Trend line Map for 2003 episode

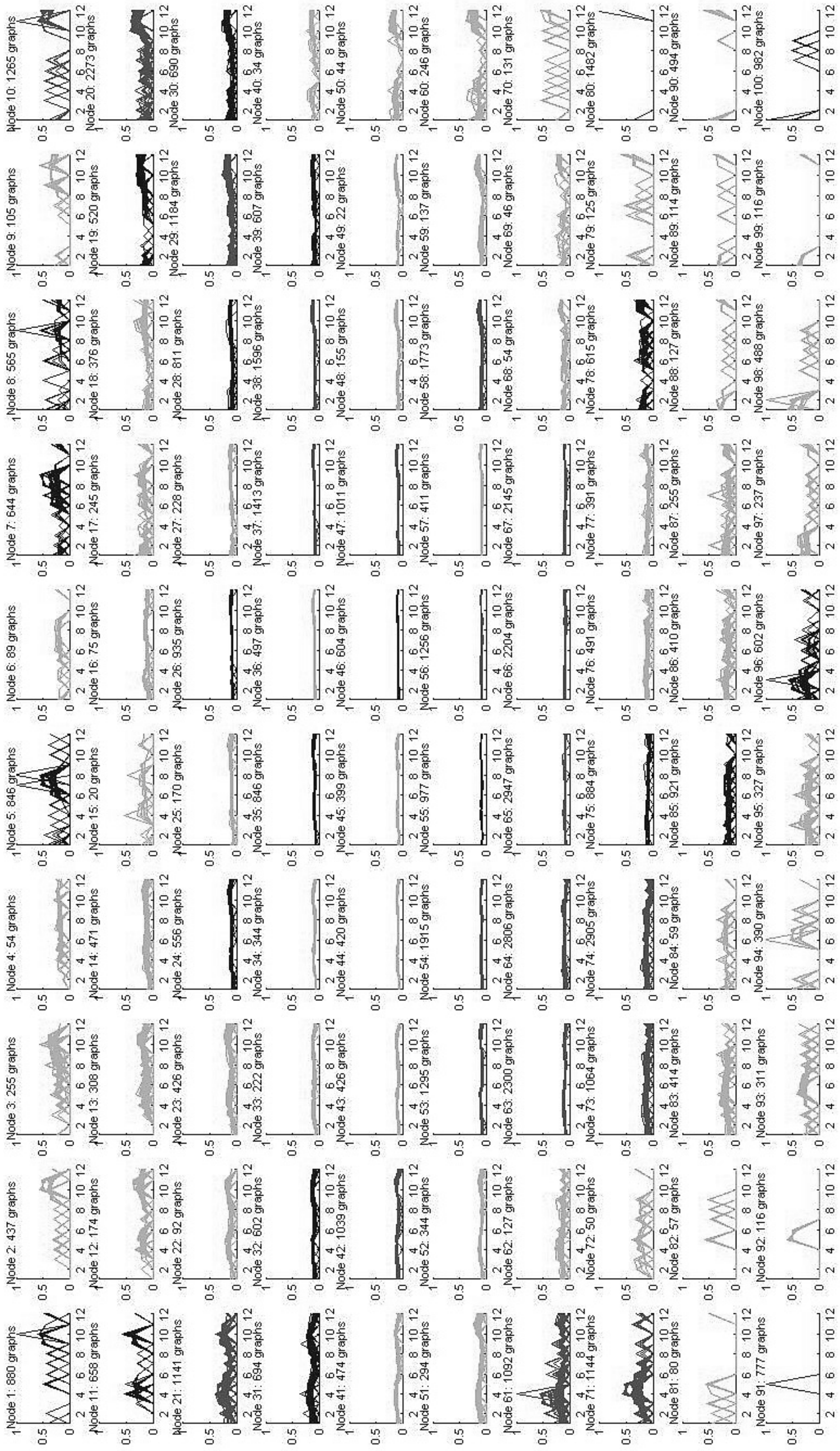


Figure 5.7: CTS network Trend line Map for 2004 episode

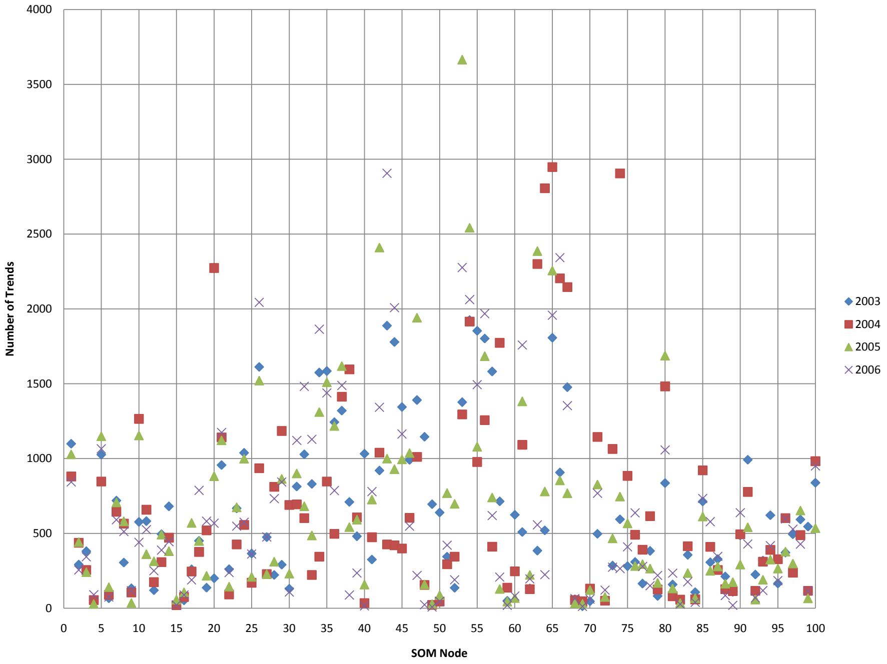


Figure 5.8: CTS network number of trends per SOM node per episode

Figure 5.8 indicates the number of trends in each cluster (i.e. cluster size) in each node for the four episodes (years) considered in this demonstration. Notice that larger quantities of patterns for all CTS episodes are grouped into the SOM nodes between 20 and 80.

5.2.2 Deeside Insurance Quotation Trend Grouping

In the experiment using the Deeside Insurance network, a 7 × 7 SOM was used and trained using the 2008 data. The prototype map is presented in Figure 5.9. From the figure it can be seen, for example, that node 1 indicates a trend line with high support mainly in February, whilst node 7 shows a trend line with high support mainly in March. It is interesting to note that there are many trends with "peaks" in the first quarter of the year (which means that there is a high probability of a request for an insurance quotation between Jan and May). However, nodes 36, 37, 38, 39, 43, 44 and 45 have a high requests for insurance quotes in September.

The prototype map was then populated with the 2008 and 2009 data to produce a sequence of two maps as shown in Figures 5.10 and 5.11. From the figures, there are a number of SOM nodes that are empty. This indicates that even though the number of trends found in the Deeside Insurance network was considerable a 7 × 7 SOM is well suited to group the Deeside Insurance trends. Notice that in the 2008 trend line map, node 30 has the highest number of trends (4239); and in the 2009 trend line map, node 24 has the highest number of trends (8851). These nodes, 30 and 24, have a steady

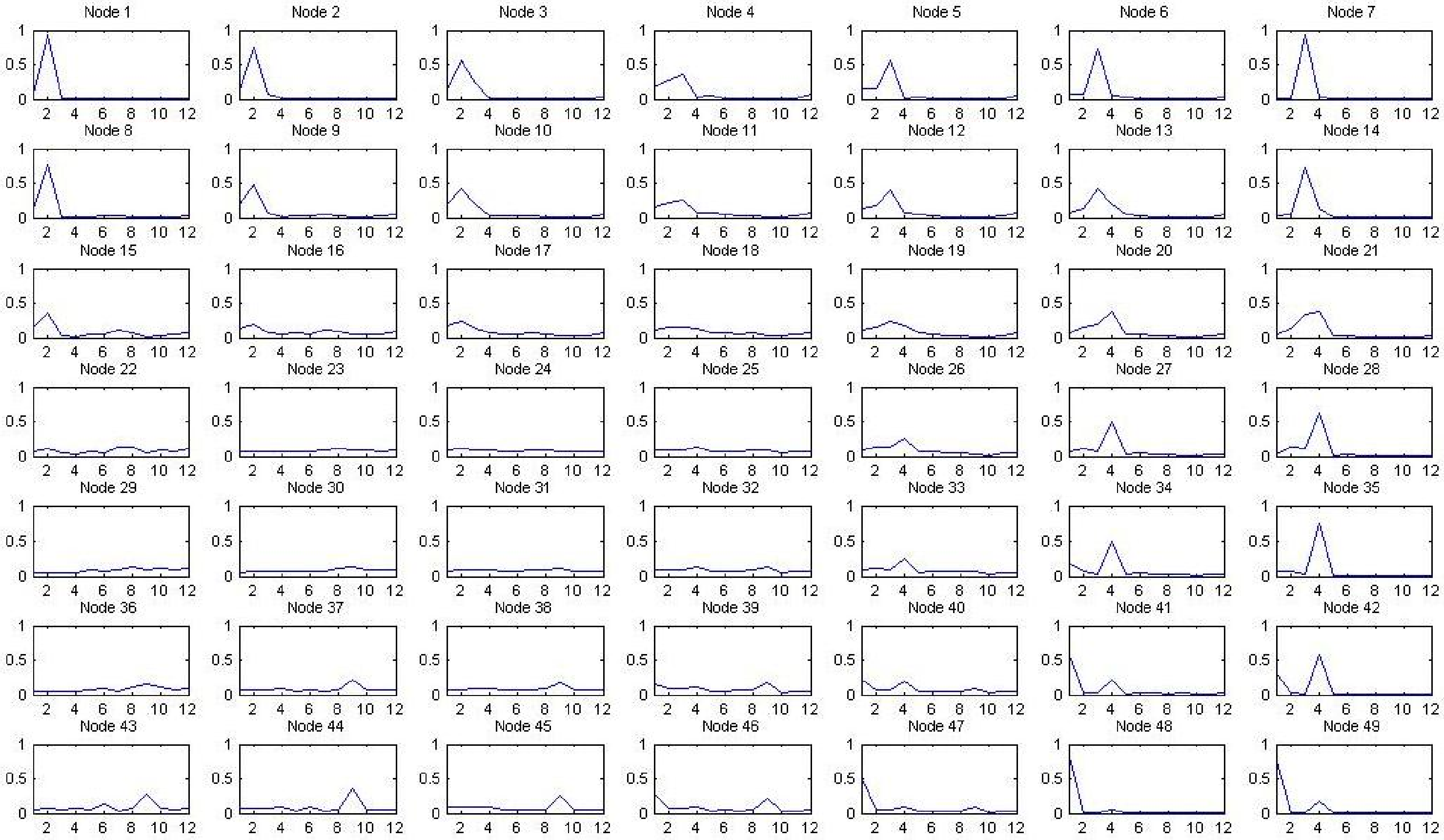


Figure 5.9: Deeside Insurance network prototype map generated using 2008 episode

trend line throughout the year.

Figure 5.12 shows the number of trends in each SOM node for the 2008 and 2009 trend line maps. A number of nodes, like nodes 1, 7, 30, 31 and 48, have a larger number of trends in the 2008 trend line map. Likewise, in the 2009 trend line map, nodes 18, 24, 29, 35 and 43, have larger numbers of trends. Using the proposed pattern migration technique it would be possible for a user to determine if there is any correlation between these sets of trends.

5.2.3 MAF Logistic Cargo Distribution Trend Grouping

To corroborate the previous findings, the MAF Logistic Cargo patterns and trends were also clustered using the Trend Grouping module. Similar to Deeside Insurance network SOM, a 7×7 node map was again used to group the trends. Figure 5.13 shows the resulting prototype map based on the trends for the 2008 data episode. From the prototype map, a number of trend types can be identified; for example, node 1 holds trend lines with high support between October and December, whereas node 34 has trend lines with support that fluctuates throughout the year.

The prototype map was then populated with the 2008 and 2009 data episodes to form the SOM trend line maps shown in Figures 5.14 and 5.15. From the maps, it can be seen that a number of nodes are empty as there are not as many MAF Logistic Cargo patterns and trends than in the case of the CTS and Deeside Insurance networks. From the experiments conducted to determine the most suitable SOM grid configuration, prototype maps of $7 \times 7, 10 \times 10$ and 12×12 were all found to provide trend line shapes that are very similar. Thus the author settled for a 7×7 map for the MAF Logistic Cargo network as the trends can still be effectively fitted to the best matching trend line according to the prototype map while gaining computational advantages over the use of the 10×10 or 12×12 grids. Inspection of both maps indicates that in 2008 the majority of trends are grouped in node 43 (494 trends), which describes trends where the cargo distribution activities are high in February; and in 2009 the majority of trends are grouped in node 22 (360 trends), which describes trends with high activity in January.

Figure 5.16 shows the overall number of trends held in the MAF Logistic Cargo 2008 and 2009 trend line maps. In the 2008 map, the nodes 1, 21, 22, 33 and 41 have larger numbers of trends in the trend clusters. Whereas in the 2009 map, the nodes 7, 21, 33, 43 and 48 have large numbers of trends. Again it would be of interest to determine whether there are any correlations between these results.

5.2.4 Experimental Analysis of Trend Grouping with Constraints

This sub-section describes the analysis of the Trend Grouping module using constraints to filter records in the CTS and Deeside Insurance networks. The analysis repeated here

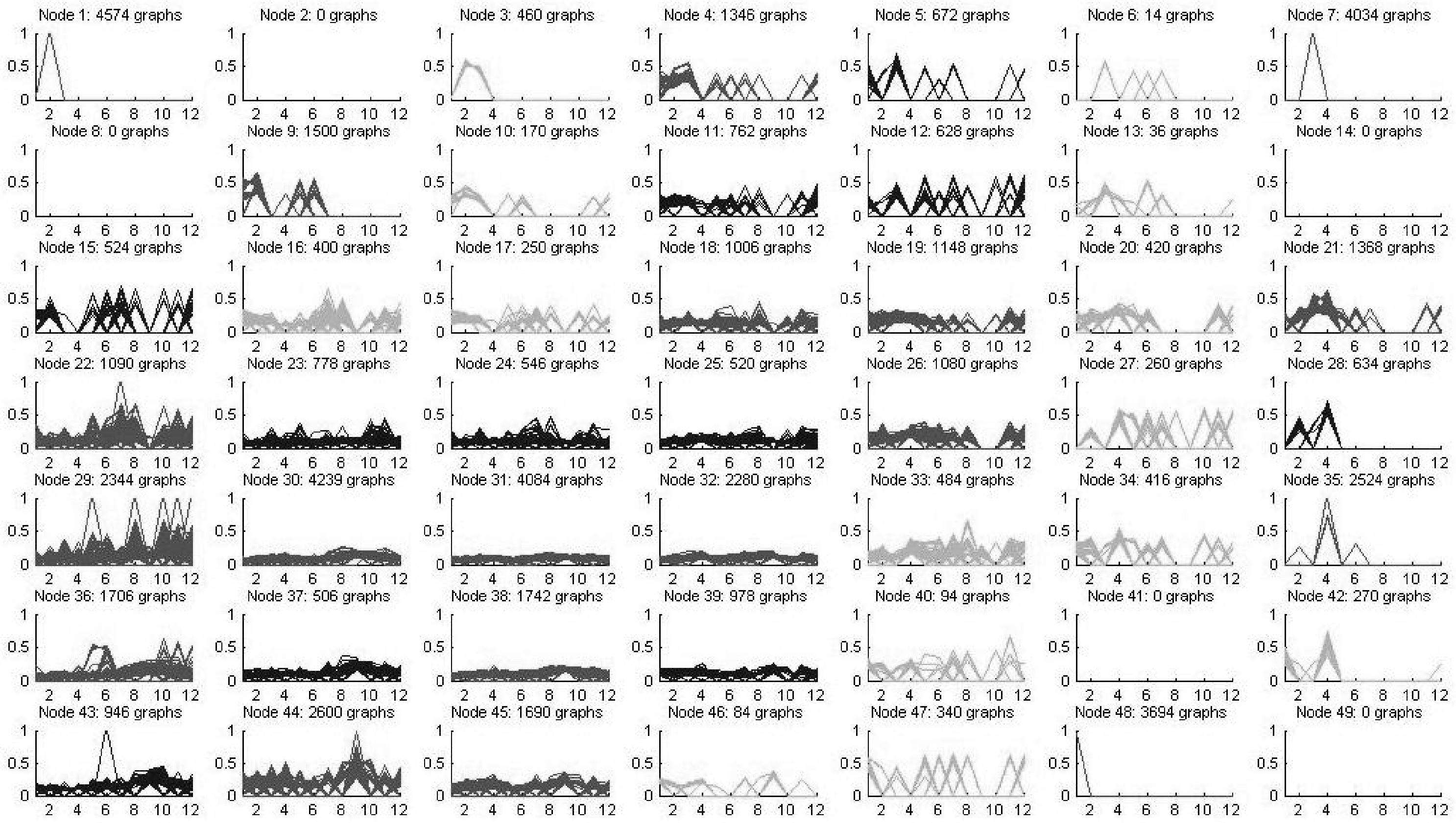


Figure 5.10: Deeside Insurance network Trend line Map for 2008 episode

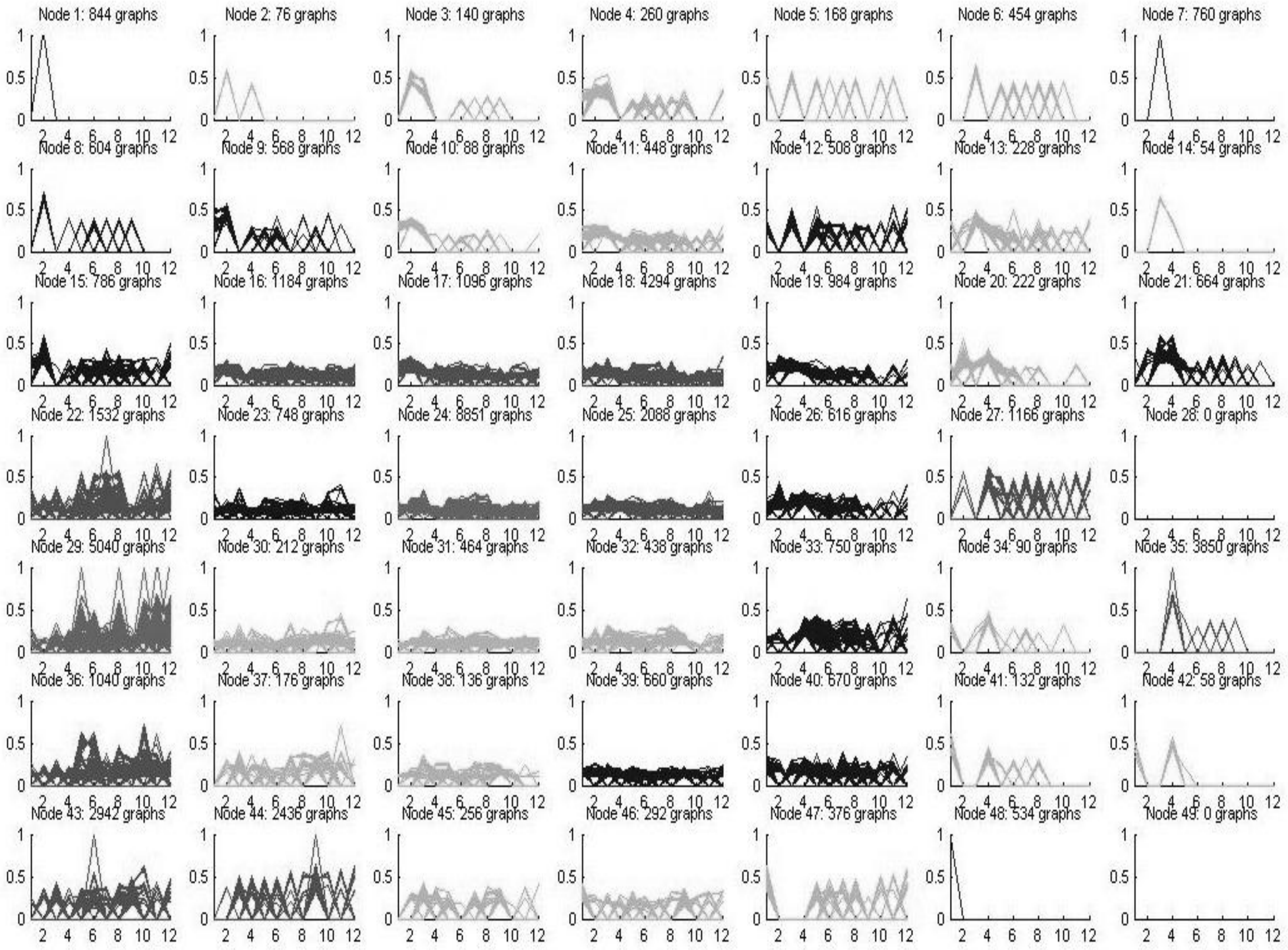


Figure 5.11: Deeside Insurance network Trend line Map for 2009 episode

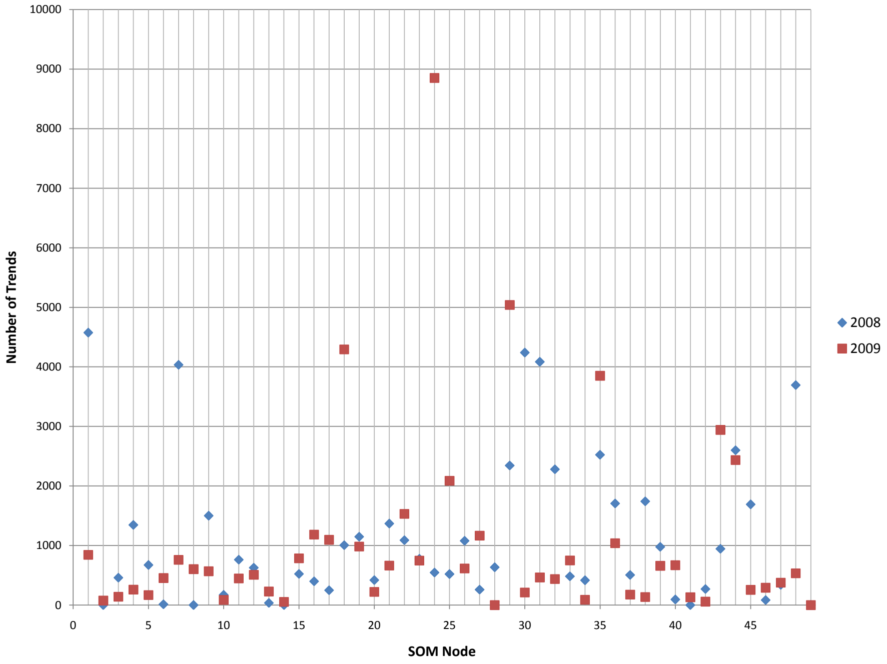


Figure 5.12: Deeside Insurance network number of trends per SOM node per episode

formed part of the experiment discussed in Sub-section 5.1.4, and which was published in [98]. For both networks, a 7×7 SOM grid was used as the number of discovered patterns and trends would be reduced.

Figure 5.17 illustrates the prototype SOM for trends generated with a support threshold of 1% with Constraint 1 ($\{Breed\ Type = Beef\}$). The prototype map displays node clusters of the discovered CTS trends. For example, node 1 (top-left) represents trends that have high support in early summer (May), while node 43 (bottom-left) indicates trend lines with high support in autumn only (October). Again, based on this prototype map, a sequence of SOM trend line maps was generated using the CTS episodes from 2003 to 2006.

Likewise, Figure 5.18 presents the prototype SOM for trends generated using a support threshold of 1% with Constraint 1 ($\{DriveAge, \{24 : 40\}\}$). The prototype map displays the characteristics of the trend line clusters. For example, with reference to the figure, node 1 (top-left) represents trends with high support from January to March, while node 18 (center) portrays trends with fluctuating support values in April, June and August. Note that, the distance between nodes indicates the dissimilarity between nodes; the greatest dissimilarity is thus between nodes at opposite ends of the diagonals. Based on this prototype map, a sequence of SOM trend line maps were generated using Deeside Insurance episodes 2008 and 2009.

Both prototype maps feature different trend line shapes, than those in the prototype maps generated without the use of constraints, indicating that the identified subset

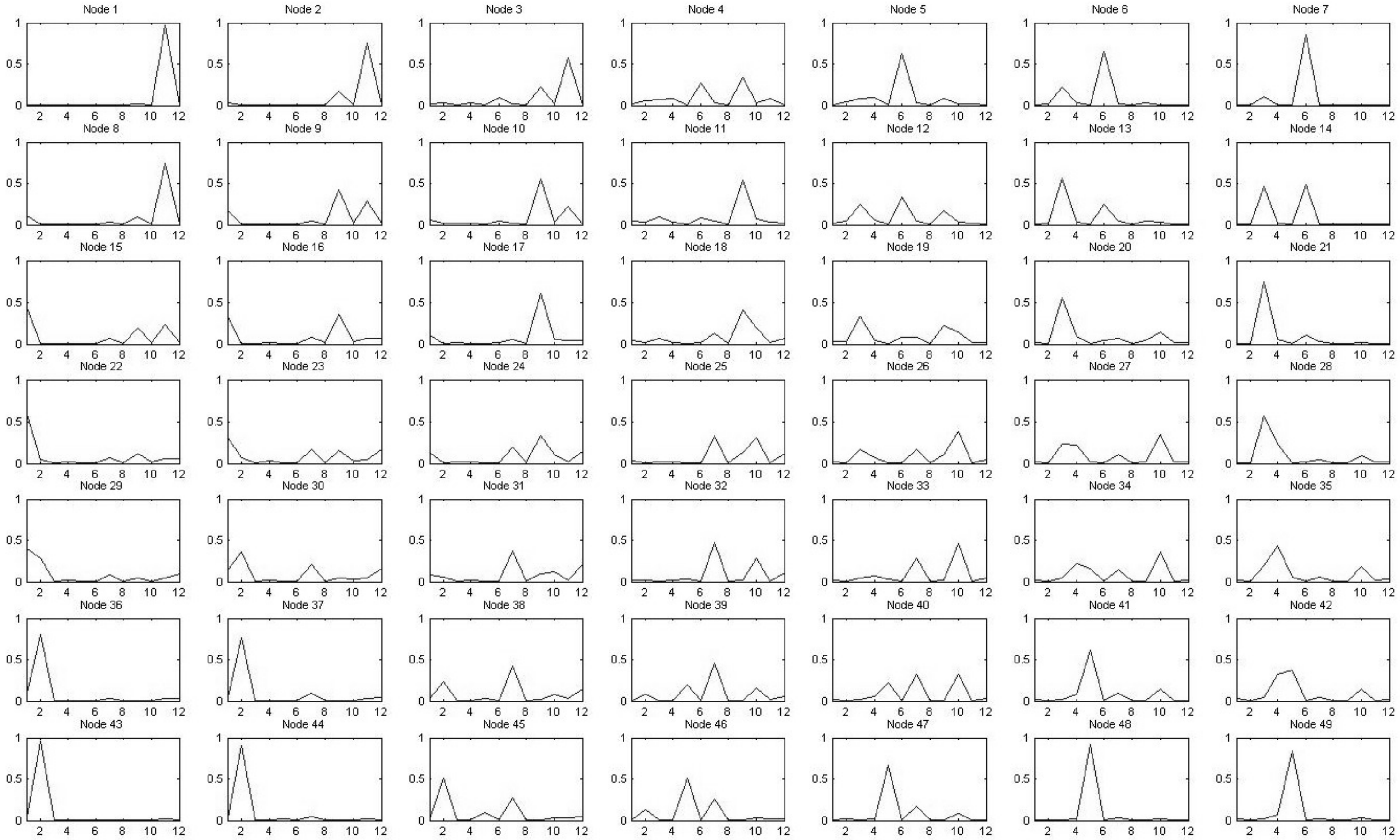


Figure 5.13: MAF Logistic Cargo network prototype map generated using 2008 episode

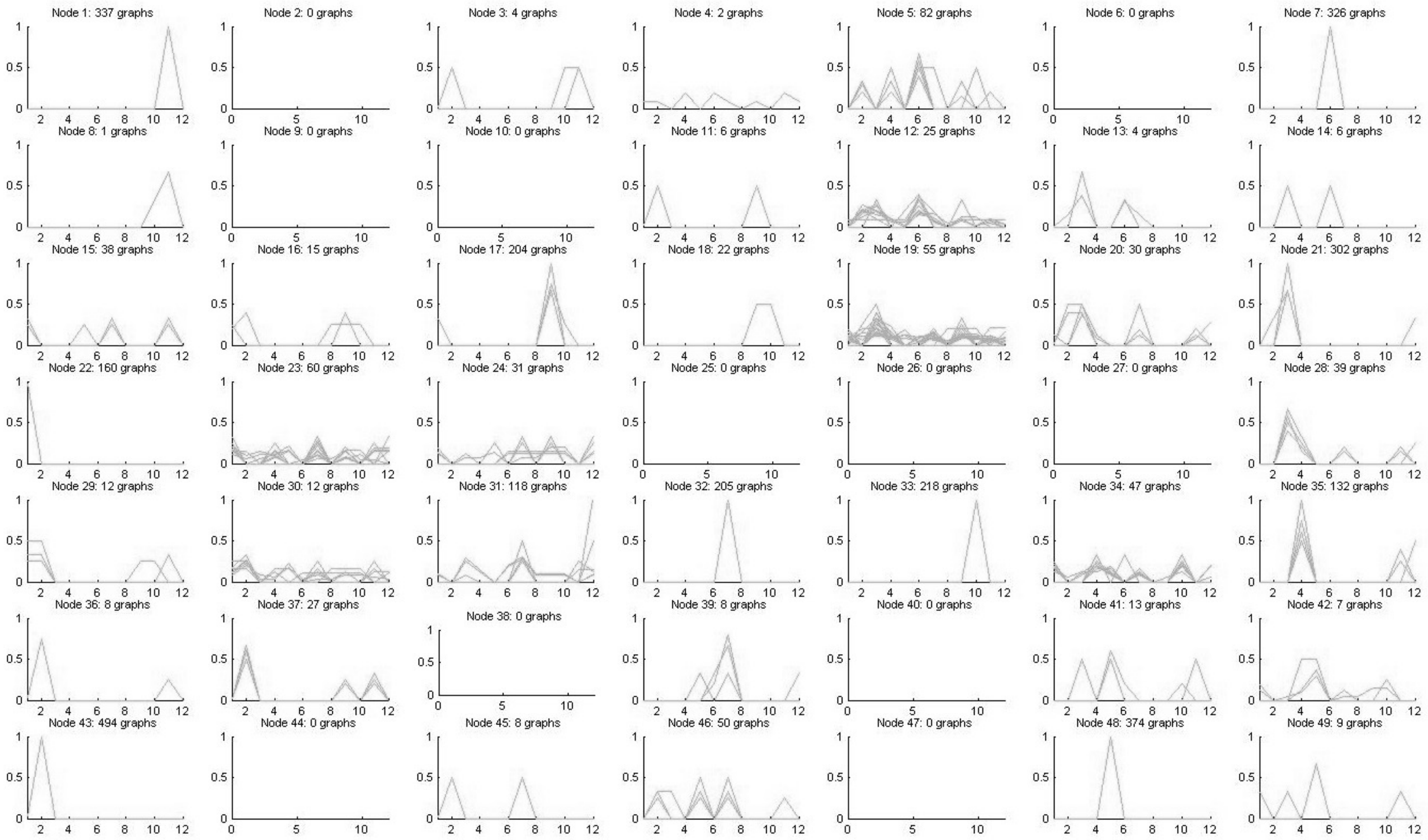


Figure 5.14: MAF Logistic Cargo network Trend line Map for 2008 episode

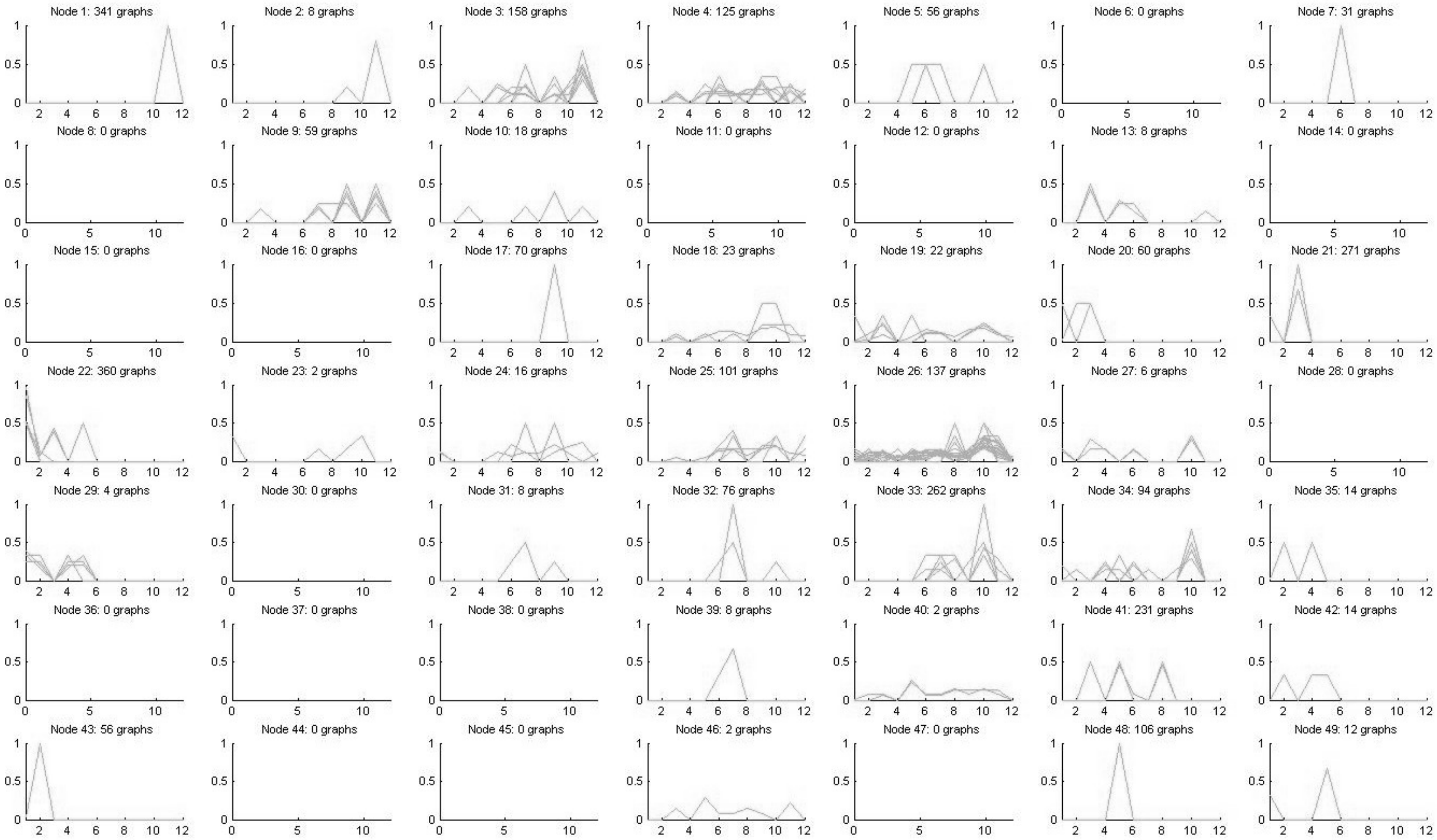


Figure 5.15: MAF Logistic Cargo network Trend line Map for 2009 episode

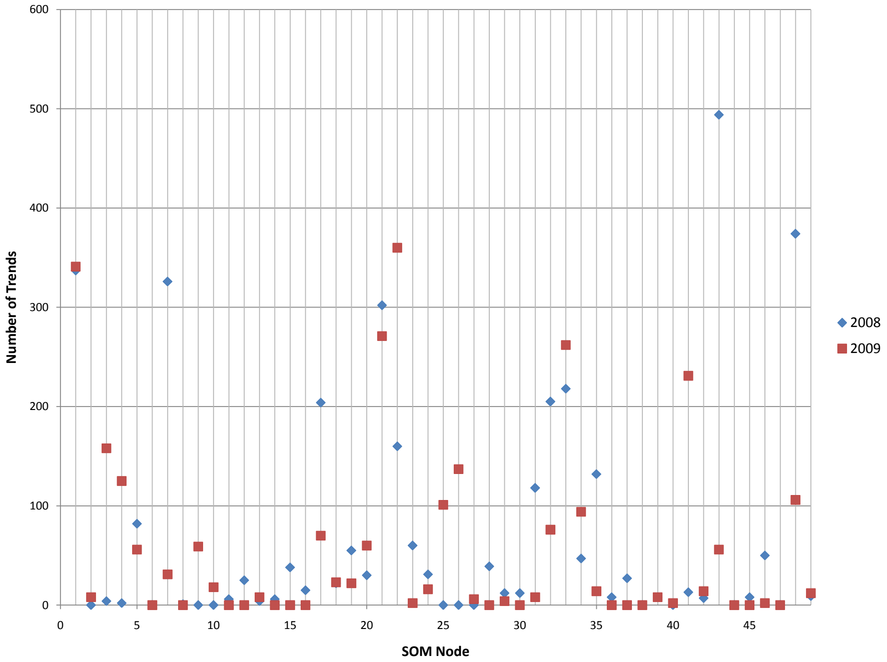


Figure 5.16: MAF Logistic Cargo network number of trends per SOM node per episode

of frequent pattern trends that are identified using constraints are different to those discovered using the entire dataset. Given the nature of the constraints used for the experiments this is to be expected. Notice also that there are not as many distinct shapes in Figures 5.17 and 5.18 as in Figures 5.5 and 5.9.

5.2.5 Trend Grouping Summary

Using the Trend Grouping module, large numbers of trends may be grouped using a set of SOM maps. This mechanism allows users to discover what types of trends are associated with the identified frequent patterns. From Figure 5.8, it could be seen that there is a great deal of variation in the size of the identified trend clusters in the CTS network data, and that consequently additional analytical support is desirable. Also, in Figures 5.12 and 5.16, large numbers of trends can be observed when using the Deeside Insurance and MAF Logistic Cargo data. Although it is argued here that the use of the concept of trend clusters provides support for the analysis of identified trends, the large number of trends that may be held within an individual cluster makes further support for enhanced analysis desirable.

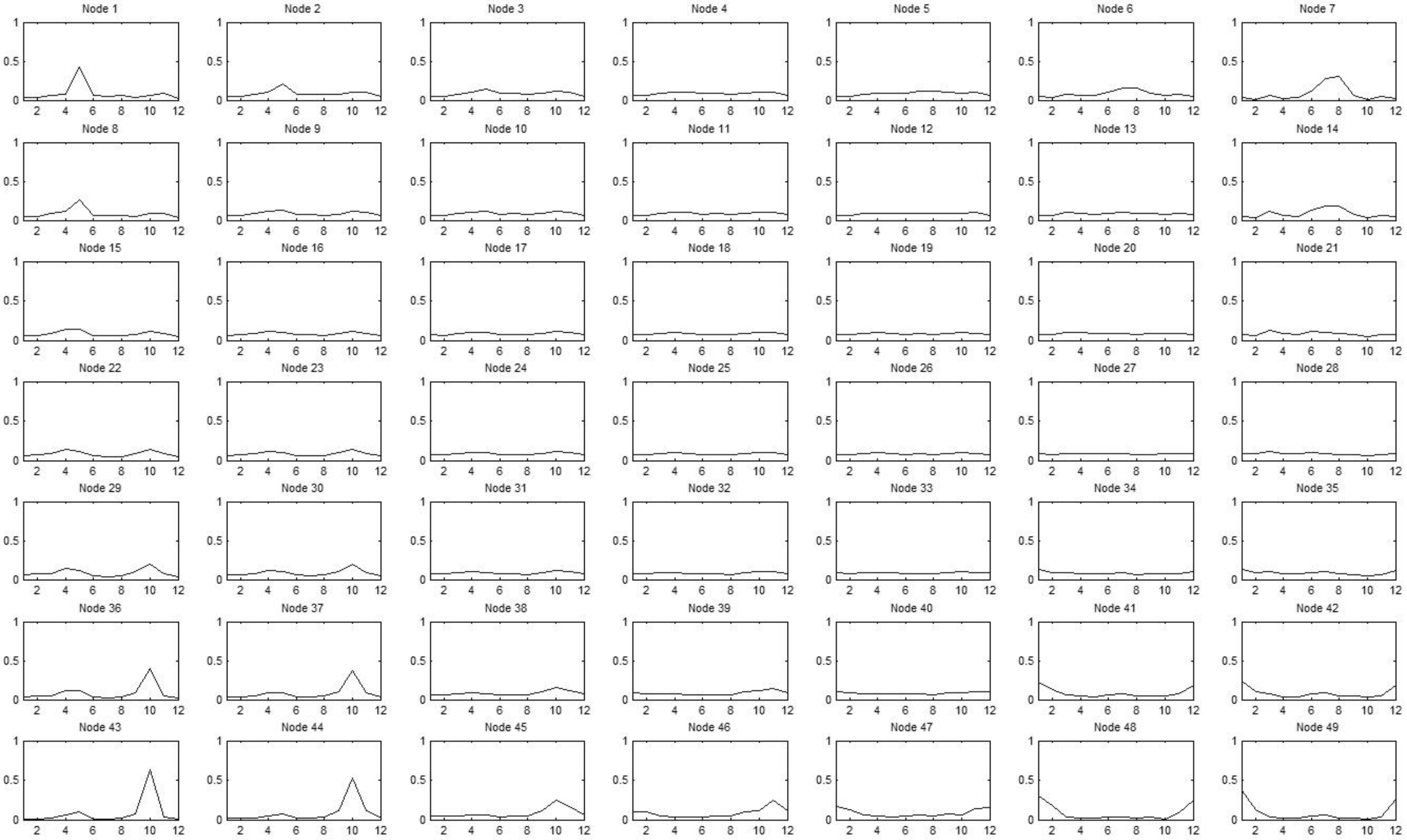


Figure 5.17: CTS network prototype map with Constraint1

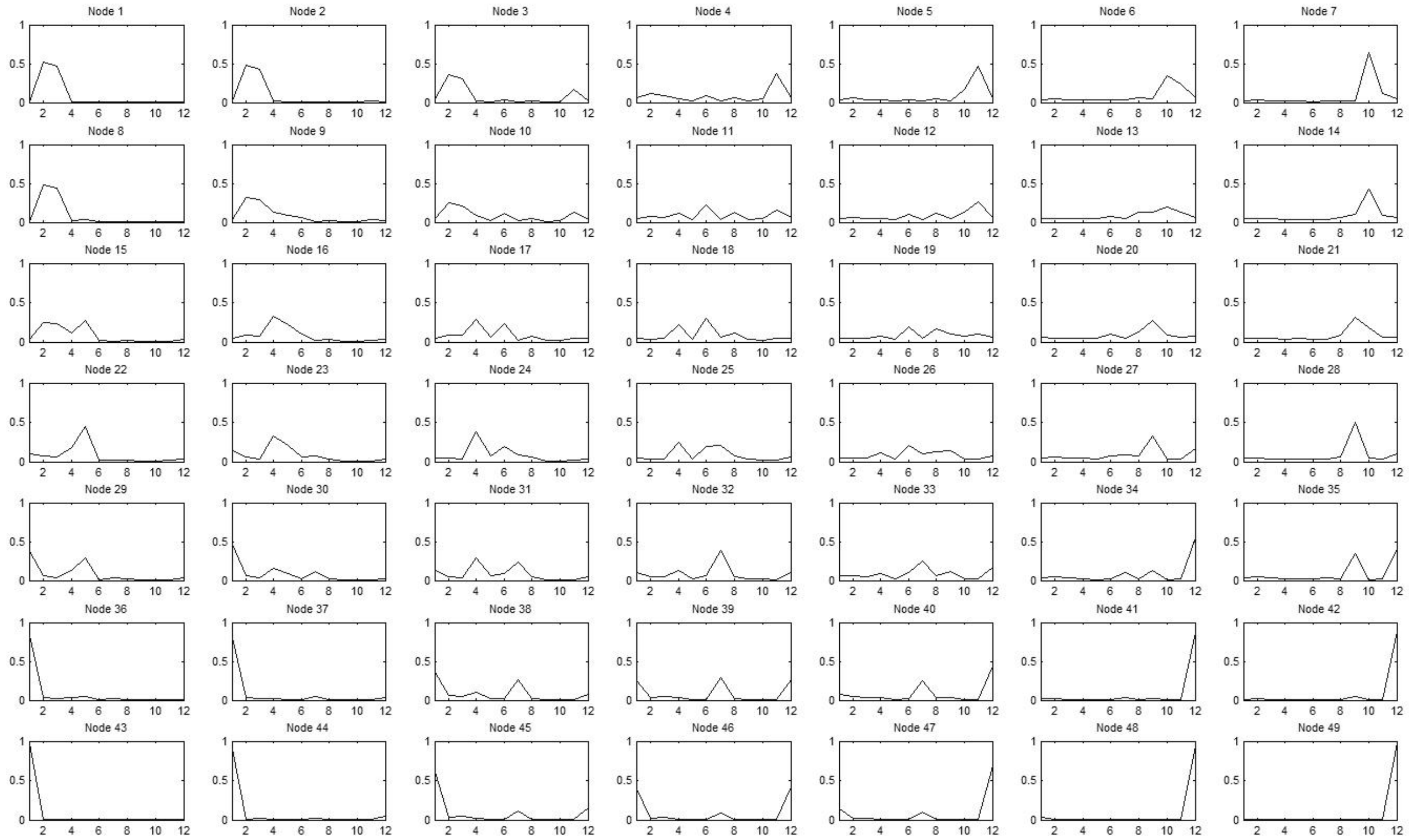


Figure 5.18: Deeside Insurance network prototype map for trends with Constraint1

5.3 Experimental Analysis of The Pattern Migration Clustering Module

In this section the analysis of the Pattern Migration Clustering module is discussed. The module groups frequently occurring pattern migrations. There are two main processes in this module, (i) identification of pattern migrations and (ii) clustering of the identified pattern migrations to determine communities of trends. It is conjectured that if there are a large number of patterns at a node n_1 in SOM M_e which then migrate to a node n_2 in SOM M_{e+1} this will be of interest. If a SOM map is viewed as a network where all nodes are linked to all other nodes, then the links represent potential "migration paths", in which case it will be useful to identify groups of nodes with high connectivity. Such groups are referred to as *islands* so as to distinguish them from the trend clusters represented by individual SOM nodes. In the context of more traditional social network analysis we might refer to these islands a *communities*. As described in Chapter 4 a hierarchical clustering method was used to group highly connected trends (SOM nodes) into islands. Once the islands have been discovered the Pattern Migration Visualisation module was used to present this information to users (see Section 5.4).

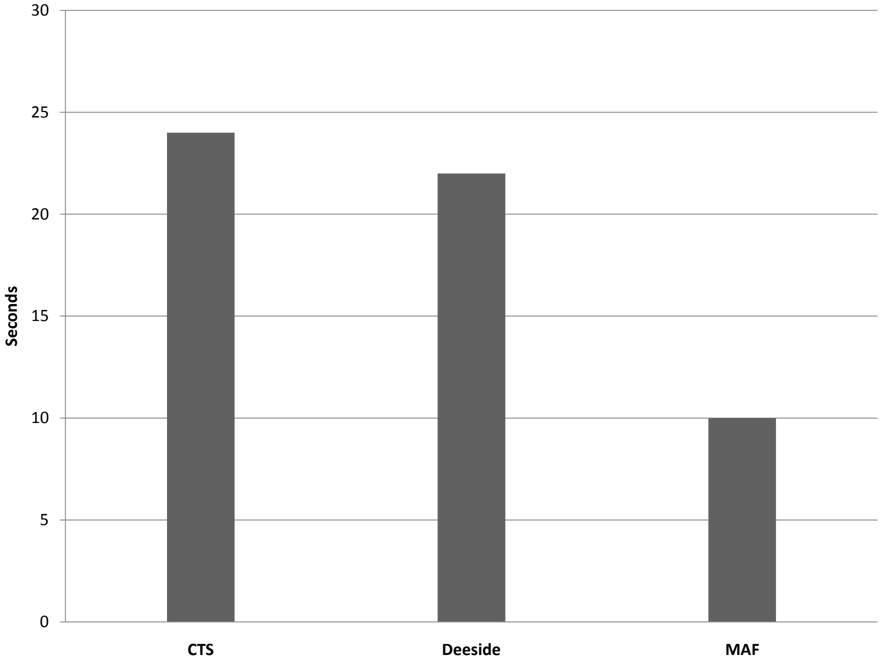


Figure 5.19: Pattern Migration Clustering module run time (seconds) for the CTS, Deeside Insurance and MAF Logistic Cargo networks

In this section each of the three network datasets is considered in turn, CTS frequent patterns and trends using a 0.5% support threshold, Deeside Insurance frequent patterns and trends using a 5% support threshold and MAF Logistic Cargo frequent patterns and trends using a 5% support threshold. Figure 5.19 shows the computational run time of the Pattern Migration Clustering Module. Processing of the CTS network required more time as it contained more patterns and trend clusters than the Deeside Insurance and MAF Logistic Cargo datasets. The number of pattern migrations in each case is presented using a Migration Matrix of the form shown in Table 5.12. The matrix shows the numbers of patterns that have migrated from M_e to M_{e+1}, $n_{i,i}$ gives the number of patterns that have stayed in cluster c_i in both trend line maps (the term *self-link* is used to indicate such migrations), $n_{i,j}$ gives the number of patterns that have migrated from c_i in M_e to c_j in M_{e+1}. The Q values required for the hierarchical clustering are calculated using these numbers of pattern migrations, and are used to cluster the pattern migrations.

	C_1	C_2	C_3	C_4	$\dots$	C_n
C_1	$n_{1,1}$	$n_{1,2}$	$n_{1,3}$	$n_{1,4}$	$\dots$	$n_{1,n}$
C_2	$n_{2,1}$	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	$\dots$	$n_{2,n}$
C_3	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$	$n_{3,4}$	$\dots$	$n_{3,n}$
C_4	$n_{4,1}$	$n_{4,2}$	$n_{4,3}$	$n_{4,4}$	$\dots$	$n_{4,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
C_n	$n_{n,1}$	$n_{n,2}$	$n_{n,3}$	$n_{n,4}$	$\dots$	$n_{n,n}$

Table 5.12: Format of a Migration Matrix

The numbers of pattern migrations (given in Table 5.12) are also used to determine the C values for the Pattern Migration Visualisation module (considered in Section 5.4). The C values (introduced in the Chapter 4) are used to identify the positions and relationships of trend cluster nodes to support the visualisation and animation of trend migrations. The analysis of the Pattern Migration Clustering module with respect to the CTS, Deeside Insurance and MAF Logistic Cargo networks are presented in Sub-sections 5.3.1, 5.3.2 and 5.3.3 respectively. Pattern migration using constraints is considered in Sub-section 5.3.4. This section is summarised in Sub-section 5.3.5.

5.3.1 GB Cattle Movement Pattern Migration

Using the trend cluster analysis algorithm (Algorithm 4.7 in Chapter 4), pattern migrations in the CTS network are identified by observing which node the pattern belonged to in M_e and where the pattern moved to in M_{e+1}. The difference between the nodes' locations in the SOM maps indicate the distance traveled. The greater the distance the more "interesting" a pattern migration may be deemed to be.

Some examples of CTS patterns migrations from one node to another between SOM M_{2003} to M_{2006} are shown in Table 5.13. From the table it can be seen that, the trend line associated with pattern $\{ReceiverArea = 14, SenderArea = 13, AnimalAge \leq 1yearold\}$ was in node 10 in the 2003 SOM map and moved to node 8 in the map for 2004, then moved to node 44 in the map for 2005 and ended up in node 18 in the map for 2006. The pattern $\{Number\ cattle\ moved\ \leq\ 5,\ Receiver\ location\ type = Slaughter\ House\ (Red\ Meat),\ Receiver\ Area = 14,\ Sender\ Area = 13, Gender = female\}$ was only considered significant (frequent) in two data episodes, 2004 and 2005. It was in node 20 in the map for 2004 and moved to node 80 in the map for 2005. Referring to the CTS network prototype map in Figure 5.5, it can be seen that the change of trend “type”, from trends that are prevalent between Sept and Dec in 2004 to trends that are prevalent from December and January in 2005, can be said to be significant (or at least interesting).

No.	Frequent Patterns	Node M_{2003}	Dist	Node M_{2004}	Dist	Node M_{2005}	Dist	Node M_{2006}
1.	$\{Receiver\ Area = 14,$ $Sender\ Area = 13,$ $Animal\ Age \leq 1yearold\}$	10	2.0	8	5.1	44	4.8	18
2.	$\{Number\ cattle\ moved \leq 5,$ $Receiver\ location$ $type = Slaughter\ House$ $(Red\ Meat), Receiver\ Area = 14,$ $Sender\ Area = 13,$ $Gender = female\}$	0	0	20	9.8	80	0	0
3.	$\{Receiver\ Area = 44,$ $Sender\ Area = 44, Breed$ $Type = Beef,\ Breed = Belgian$ $BlueCross\}$	6	1.0	5	2.8	21	5.0	38
4.	$\{Receiver\ Area = 35,$ $SenderArea = 35, Breed$ $Type = Beef\ and\ Dairy,$ $AnimalAge \leq 1yearold\}$	7	6.7	46	6.3	6	0	0
5.	$\{Receiver\ PTI = 4,$ $Receiver\ Area = 14, Sender$ $Area = 13, Gender = female\}$	33	1.4	27	5.1	29	0	29

Table 5.13: Example of migrating CTS Frequent Patterns trends

Table 5.13 also shows other patterns that migrated and experienced changes of trend types. The fourth example shows a pattern trend that was frequent from 2003 to 2005 but not frequent in 2006. Moreover, the distance values are considerably high thus signifying that this pattern migration is an interesting pattern migration.

Table 5.14 shows a fragment of the pattern Migration Matrix for the CTS network. Overall there are 100 trend clusters. The matrix contains the number of patterns that migrated from C_i in M_e to C_j in M_{e+1}. As already mentioned these numbers are used to determine the Q and C values for the proposed visualisation. The communities of CTS trends identified by the hierarchical method are shown as islands in the output

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	$\dots$	C_{100}
C_1	71	13	11	0	0	0	1	26	0	0	$\dots$	0
C_2	10	6	13	0	1	0	0	0	0	2	$\dots$	1
C_3	8	0	21	2	0	0	18	0	0	0	$\dots$	0
C_4	0	0	0	3	0	3	0	0	0	0	$\dots$	0
C_5	0	0	0	0	14	0	14	10	0	0	$\dots$	0
C_6	0	0	0	0	2	0	1	0	0	0	$\dots$	0
C_7	0	0	3	0	23	4	67	6	0	1	$\dots$	2
C_8	0	2	0	0	1	2	6	5	3	0	$\dots$	0
C_9	0	1	0	0	0	0	3	0	0	1	$\dots$	0
C_{10}	0	3	0	1	1	0	0	1	0	7	$\dots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
C_{100}	0	0	0	0	0	0	1	0	0	0	$\dots$	7

Table 5.14: Fragment of the pattern Migration Matrix from M_{2003} node (cluster) to M_{2004} for the CTS network dataset

produced using the Pattern Migration Visualisation module.

5.3.2 Deeside Insurance Quotation Pattern Migration

With respect to the Deeside Insurance network similar pattern migrations are identified as those in the CTS dataset. Comparison of the trend clusters allowed for the identification of changes in customer “quote request” habits. Table 5.15 presents some examples of pattern migrations identified from within the Deeside Insurance dataset. For example, the trend line representing the pattern $\{Fine \leq 1000, Convict\ Code = SP, 41 \leq Driver\ Age \leq 50, 1996 \leq Car\ Year\ Manufacture \leq 2000\}$ which was in node 43 (bottom right in Figure 5.9) in M_{2008} migrated to node 11 in M_{2009}. This signifies, in this case, that the pattern has changed from a trend with high support (frequency) in September to a trend with high support in February and March. Another example is the trend line for pattern $\{Fine \leq 1000, Convict\ Code = SP, 41 \leq Driver\ Age \leq 50, 2001 \leq Car\ Year\ Manufacture \leq 2005, Aggregator = Moneysupermarket\}$, this was in node 35, trends with sharp increase of quote requests in April in M_{2008}; and then migrated to node 20 which describes trends that gradually increased from January to April in M_{2009}.

Table 5.16 presents a fragment of the pattern Migration Matrix from M_{2008} to M_{2009}. The total number of trend clusters in the matrix is 49. Recall that the values in the matrix are used with respect to the Pattern Migration Visualisation module.

5.3.3 MAF Logistic Cargo Distribution Pattern Migration

Table 5.17 presents some examples of patterns that migrated from one SOM node to another with respect to the MAF Logistic Cargo network. Thus, the pattern

No.	Frequent Patterns	Node M_{2008}	Dist	Node M_{2009}
1.	$\{Fine \leq 1000, Convict\ Code = SP,$ $41 \leq Driver\ Age \leq 50,$ $1996 \leq Car\ Year\ Manufacture \leq 2000\}$	43	5.8	11
2.	$\{Fine \leq 1000, Convict\ Code = SP,$ $41 \leq Driver\ Age \leq 50,$ $1996 \leq Car\ Year\ Manufacture \leq 2000,$ $Aggregator = Moneysupermarket\}$	44	6.3	4
3.	$\{Fine \leq 1000, Convict\ Code = SP,$ $41 \leq Driver\ Age \leq 50,$ $2001 \leq Car\ Year\ Manufacture \leq 2005\}$	36	4.2	18
4.	$\{Fine \leq 1000, Convict\ Code = SP,$ $41 \leq Driver\ Age \leq 50,$ $2001 \leq Car\ Year\ Manufacture \leq 2005,$ $Aggregator = Moneysupermarket\}$	35	2.2	20

Table 5.15: Example of migrating Deeside Insurance Frequent Patterns trends

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	$\ldots$	C_{49}
C_1	120	0	0	0	26	36	20	92	34	0	$\ldots$	0
C_2	0	0	0	0	0	0	0	0	0	0	$\ldots$	1
C_3	0	0	0	2	4	4	6	46	46	2	$\ldots$	0
C_4	70	0	16	22	4	10	40	100	36	6	$\ldots$	0
C_5	16	0	0	24	10	4	16	10	0	0	$\ldots$	0
C_6	0	0	0	0	0	0	0	0	0	0	$\ldots$	0
C_7	140	0	0	8	16	40	100	10	10	0	$\ldots$	2
C_8	0	0	0	0	0	0	0	0	0	0	$\ldots$	0
C_9	36	0	40	0	0	36	92	30	28	0	$\ldots$	0
C_{10}	0	0	0	0	0	0	0	32	104	4	$\ldots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$
C_{49}	0	0	0	0	0	0	0	0	0	0	$\ldots$	0

Table 5.16: Fragment of the pattern Migration Matrix from M_{2008} to M_{2009} for the Deeside Insurance network dataset

$\{MYR50001 \leq Shipment\ cost \leq MYR100000, Receiver = 9\ KOD, Sender\ City = Batu\ Kentonmen\}$ was in node 31, representing trends with high support in July, in M_{2008}; and moved to node 5, representing trends with high support in June, in M_{2009}. The pattern $\{Receiver\ City = Sibu, Sender\ City = Batu\ Kentonmen, Sender = 91DPO\}$ was in node 34, which represents a fluctuating trend with high support in April, July and October in M_{2008}; and migrated to node 4, representing trends with high support in June and September. Table 5.18 presents a fragment of the pattern Migration Matrix for the data in the MAF Logistic Cargo network (the total number of trend clusters in the matrix is 49).

No.	Frequent Patterns	Node M_{2008}	Dist	Node M_{2009}
1.	$\{Receiver\ City = Kuching, Receiver = 5\ KOD,$ $Sender\ City = Batu\ Kentonmen, Sender = 91DPO,$ $Logistic\ item = Ordnance\ items\}$	28	1	21
2.	$\{MYR50001 \leq Shipment\ cost \leq MYR100000,$ $Receiver = 9\ KOD, Sender\ City = Batu\ Kentonmen\}$	31	4.5	5
3.	$\{Receiver\ City = Sibu, Sender\ City = Batu\ Kentonmen,$ $Sender = 91DPO\}$	34	4.5	4

Table 5.17: Example of migrating MAF Logistic Cargo Frequent Patterns trends

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	$\dots$	C_{49}
C_1	0	0	3	0	0	0	0	0	1	0	$\dots$	0
C_2	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_3	0	0	3	0	0	0	0	0	1	0	$\dots$	0
C_4	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_5	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_6	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_7	1	0	0	0	0	0	0	0	0	0	$\dots$	0
C_8	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_9	0	0	0	0	0	0	0	0	0	0	$\dots$	0
C_{10}	0	0	0	0	0	0	0	0	0	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
C_{49}	1	0	0	0	0	0	0	0	0	0	$\dots$	0

Table 5.18: Fragment of pattern Migration Matrix from M_{2008} to M_{2009} for the MAF Logistic Cargo network dataset

5.3.4 Experimental Analysis of Pattern Migration with Constraints

For completeness Table 5.19 presents some examples of CTS trends using Constraint 1 $\{Breed\ Type = Beef\}$, that migrated from SOM M_{2003} to M_{2006}, For example, the trend line representing the frequent pattern: $\{Receiver\ Area = 24,\ SenderLocationType = Algricultural\ holdings, Breed\ Type = Beef, Breed = Chianina\}$ was in node 47 in M_{2003} and moved to node 15 in M_{2004}, but then migrated to node 8 in M_{2005} and node 48 in M_{2006}. Chianina is an Italian breed of cattle raised mainly for beef. As noted previously the distance traveled is considered to be an indicator of "interestingness".

Table 5.20 shows examples of Deeside Insurance trends (representing frequent patterns), with Constraint 1 $DriverAge = \{24 : 40\}$, that have migrated from SOM M_{2008} to M_{2009}. For example, the trend line representing the pattern: $\{EngineSize = \{\leq 1000\}, CarType = Nissan, DriverAge = \{24 : 40\}\}$ was in cluster node 5 (middle top in Figure 5.18) in M_{2008}, but moved diagonally to node 49 in M_{2009}. The trend shape

Frequent Patterns	Node M_{2003}	$Dist$	Node M_{2004}	$Dist$	Node M_{2005}	$Dist$	Node M_{2006}
$\{ReceiverArea = 24,$ $SenderLocationType = Algricultural$ $holdings, Breed = Chianina\}$	47	5.66	15	1.0	8	7.07	48
$\{ReceiverArea = 24,$ $SenderLocationType = Algricultural$ $holdings, Breed = LinconRed\}$	26	0.0	26	4.24	2	2.82	18
$\{ReceiverArea = 24,$ $SenderLocationType = Algricultural$ $holdings, BreedType = Beef\}$	26	0.0	26	3.60	9	3.60	26
$\{ReceiverArea = 24,$ $SenderLocationType = Algricultural$ $holdings, BreedType = Beef,$ $Breed = Chianina\}$	47	5.66	15	1.0	8	7.07	48

Table 5.19: Examples of migrating CTS trends with constraints ($Dist$ = distance value)

Frequent Patterns	Node M_{2008}	$Dist$	Node M_{2009}
$\{EngineSize = \{\leq 1000\},$ $CarType = Toyota\}$	49	2.0	35
$\{EngineSize = \{\leq 1000\}, CarType = Toyota,$ $DriverAge = \{26 : 30\}\}$	49	3.0	28
$\{EngineSize = \{\leq 1000\},$ $CarType = Nissan\}$	26	1.41	34
$\{EngineSize = \{\leq 1000\}, CarType = Nissan,$ $DriverAge = \{24 : 40\}\}$	5	6.32	49

Table 5.20: Examples of migrating Deeside Insurance trends with constraints ($Dist$ = distance value)

has changed significantly so it may be marked as an interesting pattern.

5.3.5 Pattern Migration Clustering Summary

From the above reported evaluation it can be seen that the Pattern Migration Clustering module can detect changes in trend clusters and migrations of frequent pattern trends in a given sequence of SOM trend line maps. Even though it is possible to observe how frequent pattern trends migrate, the interpretation of the Migration Matrices is still difficult. The intuition is that the proposed visualisation module can provide further analytic support. The evaluation of this module is presented in the next section.

5.4 Experimental Analysis of The Pattern Migration Visualisation Module

Recall that the Pattern Migration Visualisation module is aimed at illustrating the frequent pattern migrations that occur in a pair of SOM maps. The visualisation comprises a node and link network map where the nodes represent clusters and the links represent migrations between clusters. The direction of a link between a pair of trend cluster nodes shows a migration from a node in M_e to a node in M_{e+1}. The size of a node in a network map indicates the number of patterns at the node. The evaluation of this module used the output of the three sets of frequent patterns and trends from the CTS, Deeside Insurance and MAF Logistic Cargo networks used in the evaluation of the Pattern Migration Clustering module, namely: (i) CTS patterns and trends using a 0.5% support threshold, (ii) Deeside Insurance patterns and trends using a 5% support threshold and (iii) MAF Logistic Cargo patterns and trends using a 5% support threshold.

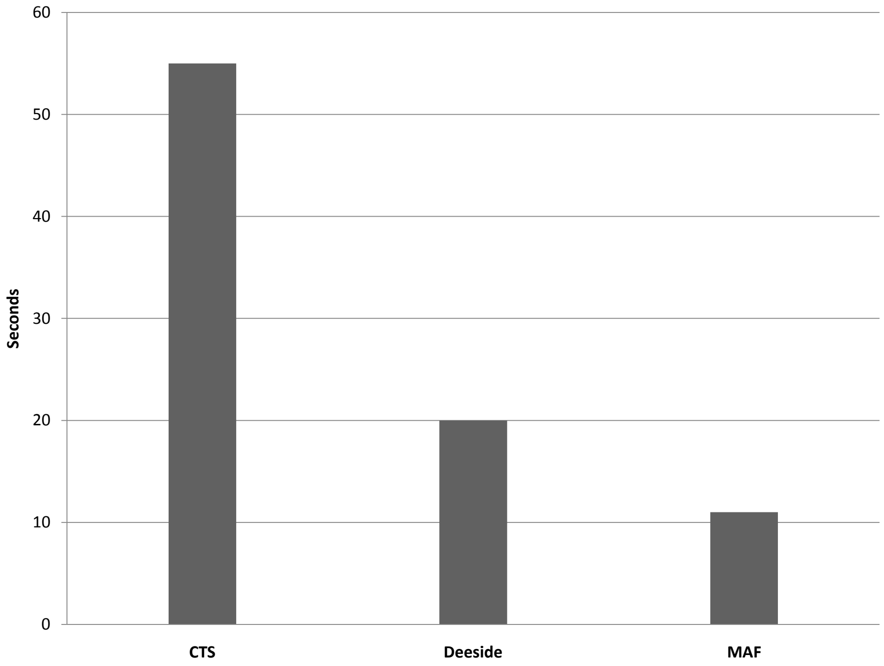


Figure 5.20: Pattern Migration Visualisation module run time (seconds) for the CTS, Deeside Insurance and MAF Logistic Cargo networks

Figure 5.20 shows the computational run time required by the Pattern Migration Visualisation module. The processing of the CTS network required a longer time as the number of patterns and trend clusters is greater than for the Deeside Insurance and MAF Logistic Cargo networks. Moreover, CTS has four episodes compared to the other networks which only have two episodes. The analyses are described in turn in the following sub-sections. Note that the experiments directed at the Pattern Migration Visualisation module did not include experiments using constraints. The section is summarised in Sub-section 5.4.4.

5.4.1 GB Cattle Movement Pattern Migration Visualisation and Animation

With respect to the CTS application, the domain users may be particularly interested in how patterns and trends change with time (from one episode to the next) because the movement of cattle is a factor with respect to the spread of bovine diseases. Using the information presented in Table 5.14 the proposed extension of Visuset was used to generate the network maps shown in Figures 5.21 and 5.22. These maps feature islands of trend clusters which can be viewed as communities in the network. This is determined using the Q values that are calculated using information of the form given in Table 5.14. The migration of CTS patterns from: episode 2003 to 2004, episode 2004 to 2005 and episode 2005 to 2006; are shown in Figures 5.21, 5.22 and 5.23 respectively. In all cases the Min-Rel threshold was set to 0.2.

Inspection of Figure 5.21 shows that the visualisation displays 45 nodes out of a maximum of 100, thus only 45 nodes included links with a C-value greater than 0.2 (and are therefore deemed interesting). The circular pattern in which the nodes are arranged on completion of the spring model algorithm is typical of the display produced (initially all nodes are placed along a diagonal). Several islands are displayed, determined using the Newman method described previously, including a large island comprising eight nodes. The nodes are annotated with an identifier (the “from” SOM node number) and the arcs with their C-value number. From the map, there are a relatively large number, 30 in all, of self-links; excluding self-links there are only 18 links indicating that, with respect to the 2003 and 2004 episodes, the patterns are fairly constant. However, the map does illustrate that (for example) patterns have migrated from node 34 to node 44, and from node 44 to 54. Referring back to Figure 5.5, an observation can be made that the nodes hold a fairly similar shape of trend line which have consistent numbers of cattle movements throughout the 12 month time stamps.

Figure 5.22 shows the migration of patterns from episode 2004 to episode 2005. Comparing this map with the previous, 2003-2004 map, it can be noted that more “islands” have appeared indicating more pattern migration communities. For example, comparison of the maps shows that, whereas between 2003 and 2004 patterns were migrating from node 44 to 54, in 2004 to 2005 there was no such migration. To give one more example, in the 2003 and 2004 map, patterns migrated from node 31 to 21; then in 2004 to 2005 they moved back from node 21 to 31. It should also be noted that node 34 is not displayed in the 2004-2005 map because the C-values for its associated links are all below the Min-Rel threshold value of 0.2 (in the 2003-2004 map the C-

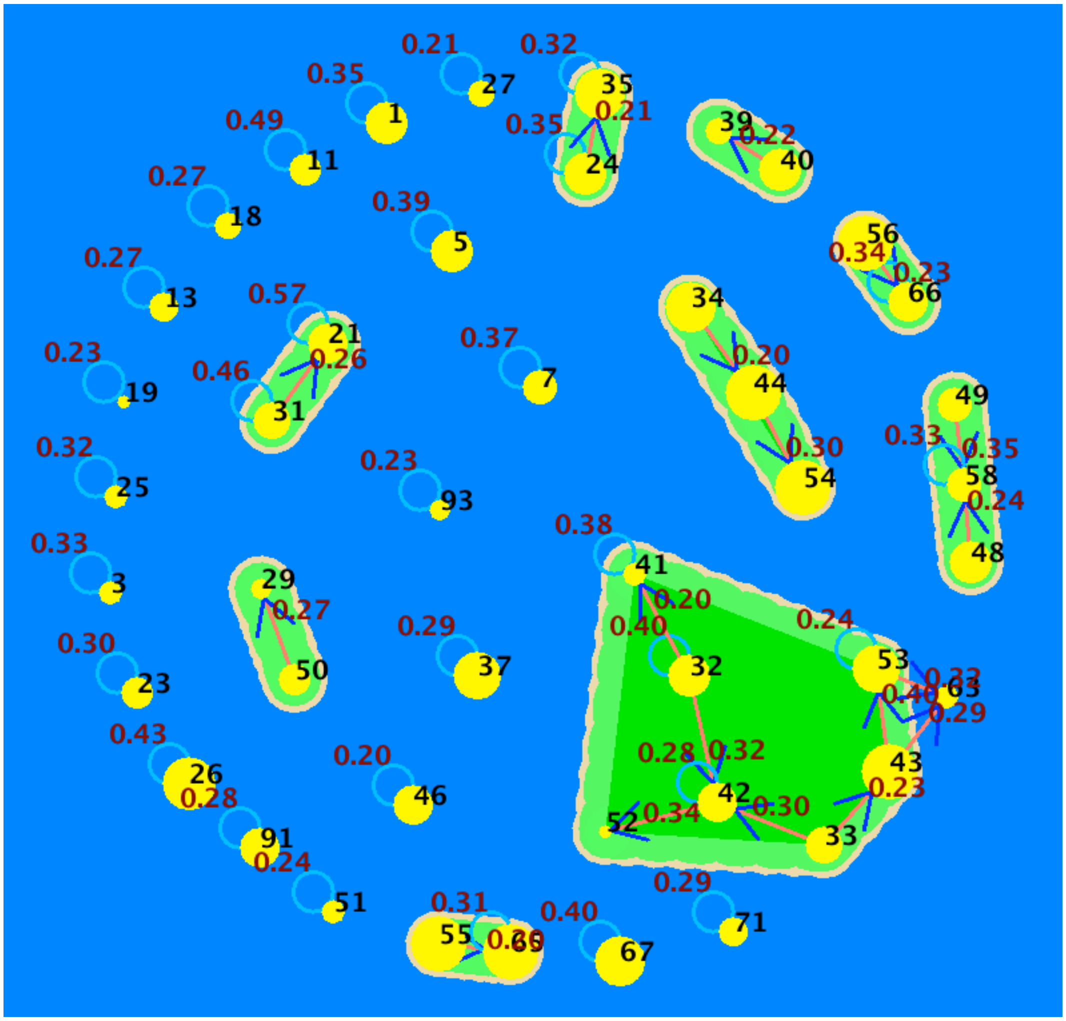


Figure 5.21: Visuset visualisation (map) indicating migration of CTS patterns from episode 2003 to episode 2004

value displayed for node 34 was only 0.2 so this is not surprising). When the animation provided with Visuset was run (although this cannot be illustrated here), it showed that node 34 disappears half way through the animation, thus indicating that the C-value is about 0.19.

The visualisation of pattern migrations from 2005 to 2006 is shown in Figure 5.23. Comparing all three maps, this last map has the least number of nodes, 29 nodes in total, that have pattern migrations with C-values of above 0.2. In general, as might be expected, most of nodes in this map have appeared in the previous maps, 2003-2004 and 2004-2005. For example, in the map 2003-2004, there are migrations of patterns with respect to nodes 11, 21 and 31; but only nodes 21 and 31 formed an island whereas node 11 showed a self-link pattern migration. In the maps 2004-2005 and 2005-2006, these 3 nodes are connected showing pattern migrating between them. The difference is in the direction the patterns moved. In the map 2004-2005, patterns move from node

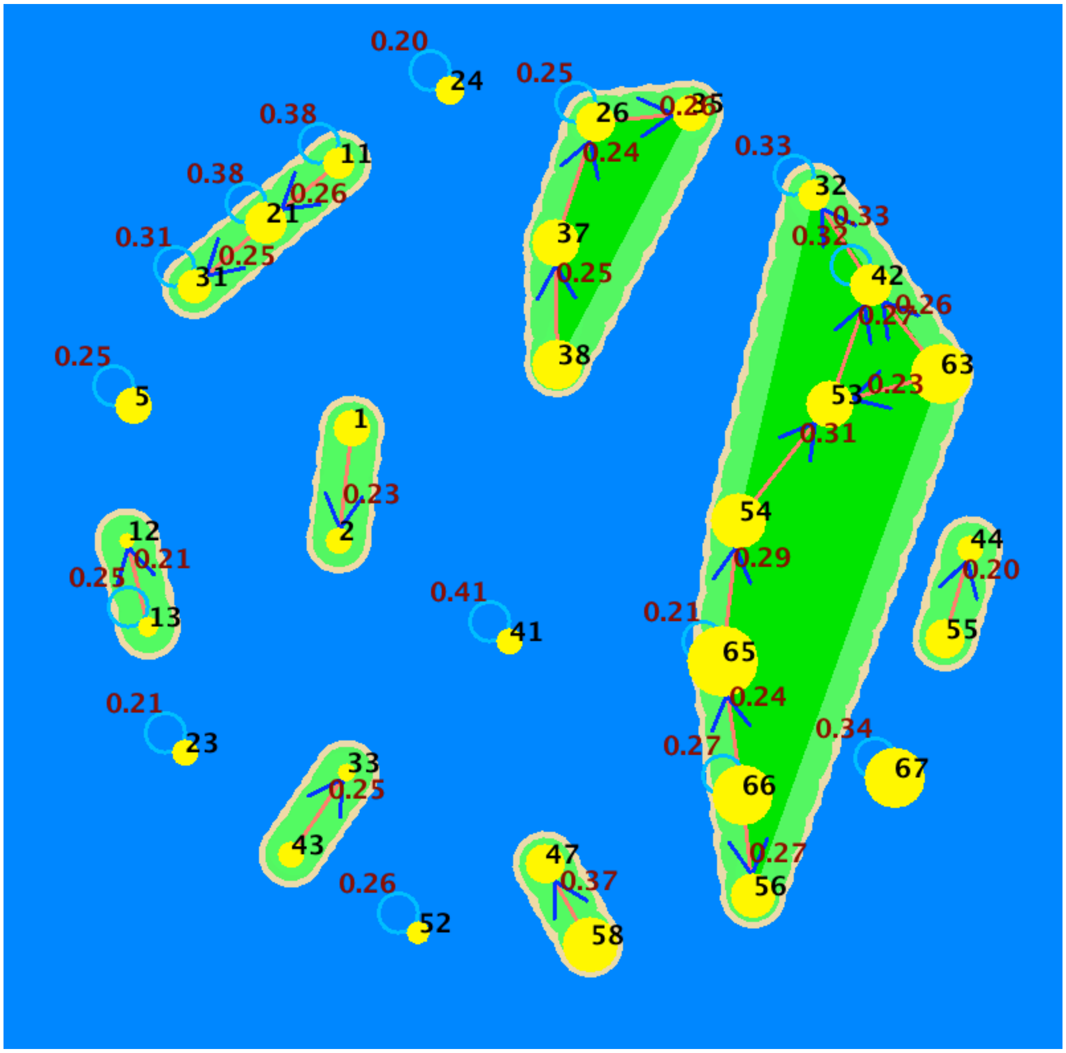


Figure 5.22: Visuset visualisation (map) indicating migration of CTS patterns from episode 2004 to episode 2005

11 to node 21 and from node 21 to node 31; whereas in the map 2005-2006, the direction of pattern migration is in the opposite direction. Additionally, node 41 is added to the island because a sufficient number of patterns moved from node 41 to node 31 in the following year.

5.4.2 Deeside Insurance Quotation Pattern Migration Visualisation and Animation

Studying the temporal change in the Deeside Insurance patterns and trends may provide useful information on how to improve Deeside Insurance's marketing strategies. In this case Visuset generates the network map using information of the form given in Table 5.16. Figure 5.24 shows the Deeside Insurance pattern migrations from 2008 to 2009. It should be recalled that the number of Deeside Insurance network data records is far

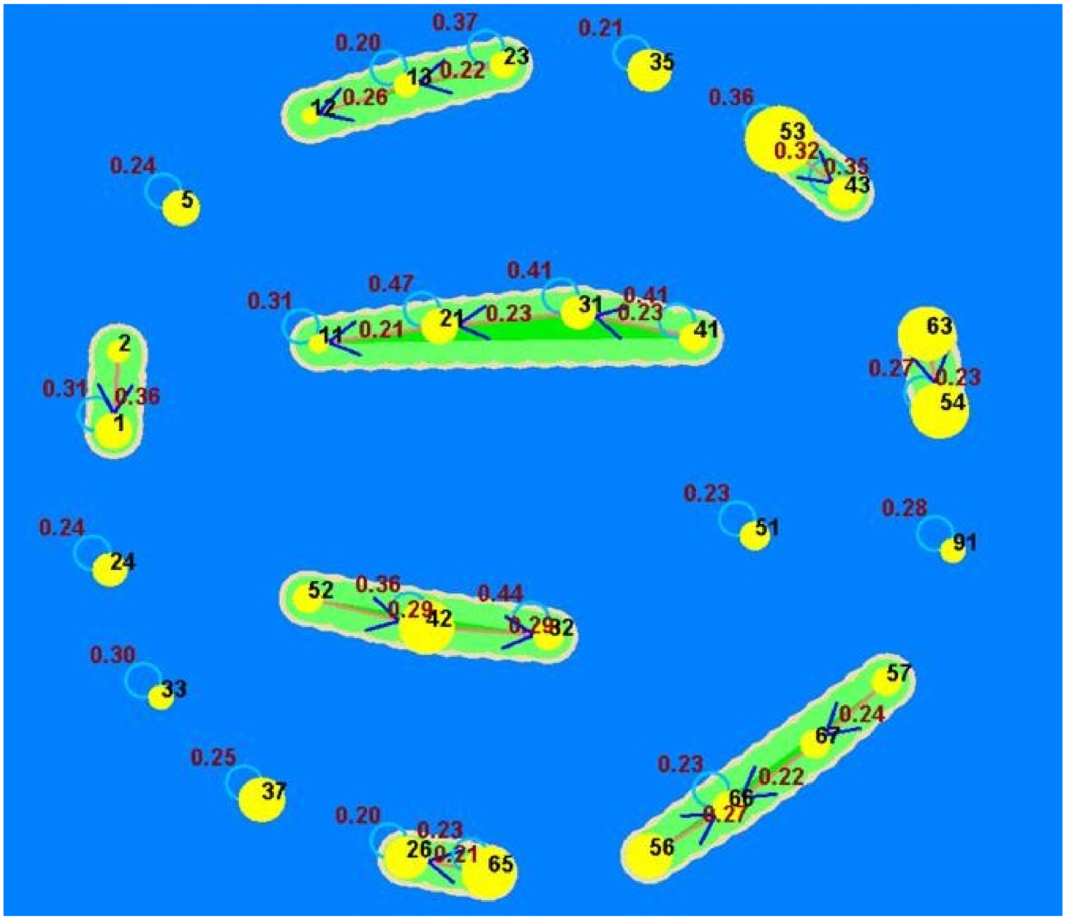


Figure 5.23: Visuset visualisation (map) indicating migration of CTS patterns from episode 2005 to episode 2006

less than for the CTS network. Therefore there are only 13 nodes, out of a total of 49, that have migration patterns with C-values of above 0.2 and thus only 5 islands of pattern migration communities are formed. Only nodes 19, 21 and 35 have self-link pattern migrations. Inspection of the data displayed in Figure 5.24 indicates that in this case the patterns migrated to similar types of trend cluster. For example, patterns from node 10 moved to node 9, with a C-value of 0.33, in the following year. Referring back to the prototype map in Figure 5.9, nodes 10 and 9 can all be categorised as trend types with high support in the first quarter of the year (between Jan to April). Also, from the map, it can be concluded that node 24 received pattern migrations from two nodes, 30 and 31, from the previous year; all three trend cluster nodes have consistent support throughout the year. The same observation can be made with respect to the rest of the islands in the 2008-2009 map, the patterns tend to migrate to adjacent trend clusters nodes (see Figure 5.9).

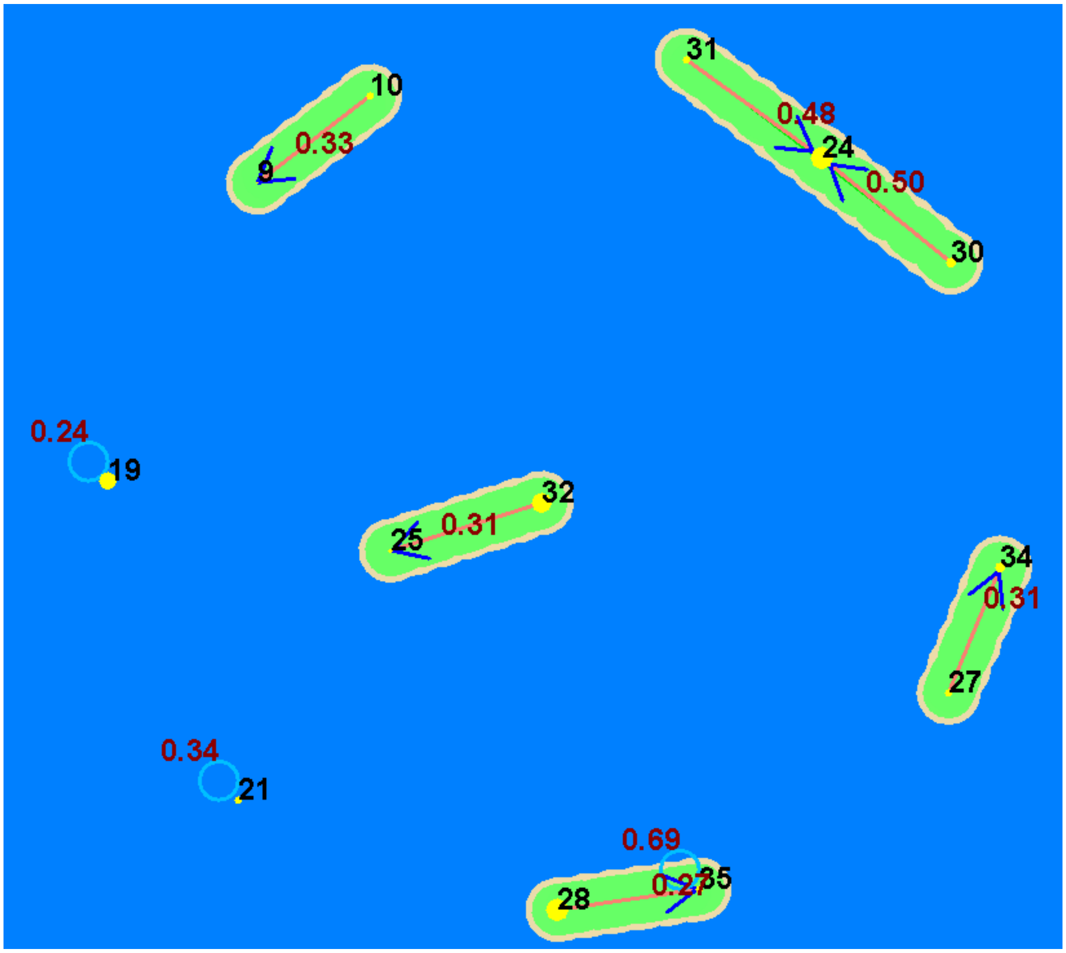


Figure 5.24: Visuset visualisation (map) indicating migration of Deeside Insurance patterns from episode 2008 to episode 2009

5.4.3 MAF Logistic Cargo Distribution Pattern Migration Visualisation and Animation

The MAF Logistic Cargo pattern migration visualisation may provide useful information for (say) inventory management and distribution scheduling. Based on Table 5.18, the Q and C values are determined to generate the network map. Figure 5.25 shows the migration of patterns from 2008 to 2009 with respect to the MAF Logistic Cargo network. From the map it can be seen there are only 25 nodes out of a total 49 cluster nodes that have pattern migrations with C-values above 0.2. None of the nodes shown in the map has a self-link pattern migration, but 7 islands of pattern migration communities have been identified. The largest island consists of 12 nodes. Inspection of the map indicates, for example, that patterns migrated from nodes 23 and 24, representing trend clusters with high support in September (refer to Figure 5.13), to node 26, a trend cluster with high support in March, Jun, July, September and October. Some patterns in node 24 also moved to node 10 which is a trend cluster with high support in September and November. The pattern migrations occurring in this island indicate distinctive changes of the trend cluster types. For example, the patterns in node 15 have a trend with high support in January, July, September and November in 2008 but migrated to node 34 with a trend of high support between April, May, July and October in 2009. Thus it could be concluded that the distribution of logistic items is not based on seasonal considerations, but may instead depend on the need or budget of MAF offices. Alternatively the migration may have occurred as a result of some change in logistic procedure/policy.

5.4.4 Pattern Migration Visualisation Summary

The Pattern Migration Visualisation module provided further analytical support to allow users to investigate pattern migrations between SOM maps that have been generated using the Trend Grouping module. Using the animation facility, the migration of patterns can be illustrated and changes of trend type associated with temporal patterns highlighted. The above discussion, focusing on the migration of particular patterns, indicates that the proposed visualisation mechanism provides a useful tool for decision makers.

5.5 Summary

In this chapter, results from a number of experiments undertaken to analyse the Frequent Pattern Trend Analysis element of the proposed framework have been reported. The analysis was considered in terms of the Trend Identification, Trend Grouping, Pattern Migration Clustering and Pattern Migration Visualisation modules. The experiments were conducted using three social networks: (i) CTS, (ii) Deeside Insurance

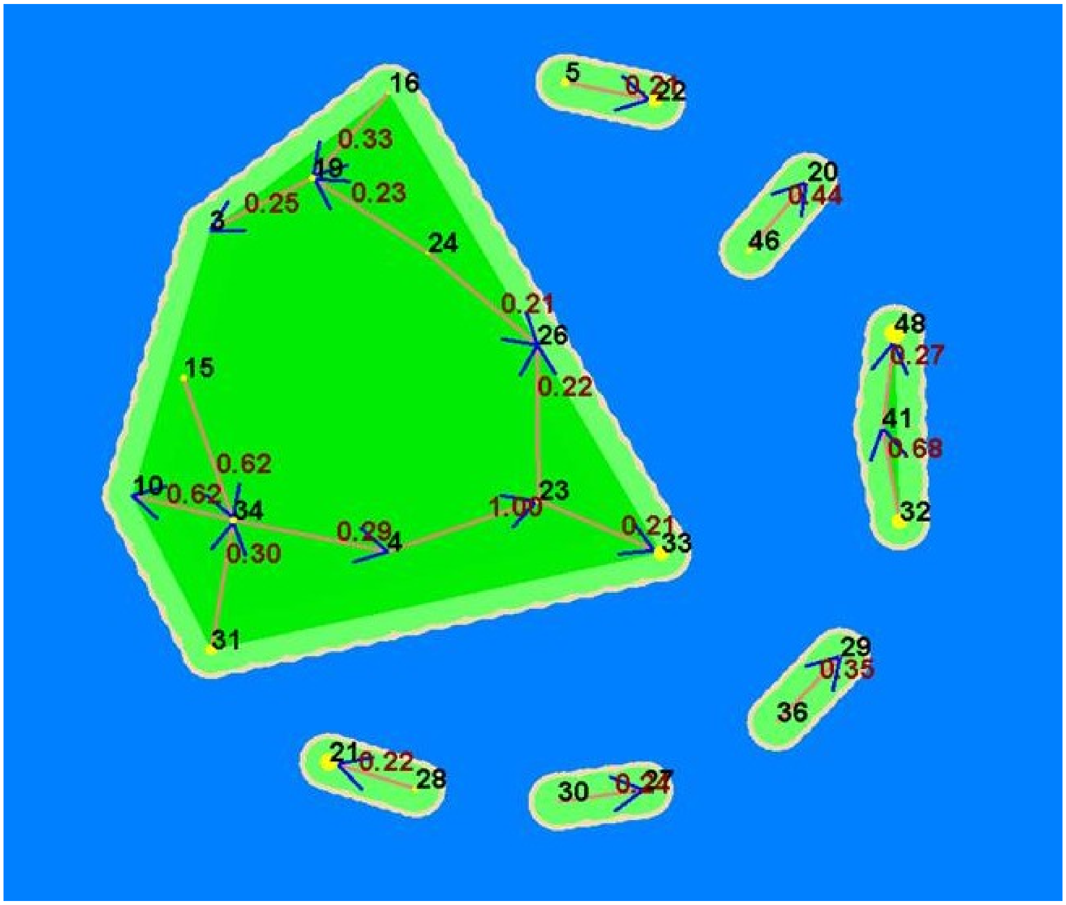


Figure 5.25: Visuset visualisation (map) indicating migration of MAF Logistic Cargo patterns from episode 2008 to episode 2009

and (iii) MAF Logistic Cargo networks.

The analysis of the Trend Identification module showed that a large number of frequent patterns and trends are discovered using the TM-TFP algorithm tending their interpretation to be difficult. The discovered trends are thus grouped using the Trend Grouping module, based on SOM technology. The module generated prototype maps and trend line maps to classify the types of trends that exist for all discovered patterns. From this experimental analysis, the Trend Identification and Grouping modules highlighted interesting information which was conjectured to be beneficial to decision makers. The proposed use of constraints further assisted decision makers in that it allowed them to "focus in" on particular types (clusters) of trends. The Pattern Migration Clustering and Pattern Migration Visualisation modules provided additional support for the analysis of the network data. The evaluations conducted with respect to the Pattern Migration Clustering module indicated that it provided for the identification of pattern migrations between trend clusters and pattern migration communities. The Pattern Migration Visualisation module then allowed users to view pattern migrations between pairs of SOM maps. The animation facility included in the visualisation module allowed for the demonstration of how trend configurations change with time.

In the following chapter, the prediction element of the proposed framework is presented to illustrate how frequent patterns (for example information or events) may be predicted to "travel" across a network. The prediction modules use the patterns and trends discovered by the Trend Identification module evaluated in this chapter.

Chapter 6

Prediction Modeling

Using modern ICT infrastructures social networks may change rapidly. The static "snap shot" node and link model of a social network describes the structure of a network and gives an indication of how information moves across the network (both directly and indirectly) at a given instance of time. However, such static analysis does not necessarily present a "true" picture. The proposed mechanisms described in the previous chapter to support dynamic analysis of networks can be argued to go some way to presenting a better picture. The work described in this chapter extends the capabilities provided by the mechanisms described in the foregoing chapter. Regardless of the type of social network under consideration (online social network, business community, file sharing systems, co-authoring framework, etc) the prediction of how an activity or event may spread across a network can clearly provide useful information with respect to many applications.

This chapter describes how the frequent patterns and trends discovered using the previous described modules may be used for prediction modeling. The work described in this chapter is motivated by a desire to use the discovered patterns and trends to predict the "percolation" of activities in networks. The work is also influenced by the concept of causal chains in networks [103, 110] which in turn suggests the use of the trends associated with identified frequent patterns as probabilistic indicators with which to determine the frequency of traffic percolating across a network.

This chapter thus presents the second part of the proposed Predictive Trend Mining Framework (PTMF). This second part comprises two modules: (i) the Percolation Matrix module and (ii) the Visualisation module. Collectively these two modules are referred to as the Prediction Modeling (PM) modules. The Percolation Matrix module operates as follows: (i) filter a set of frequent patterns of interest, and (ii) calculate the probabilities of information or events traveling from one node to another. The Visualisation module is used to illustrates the result from the Percolation Matrix module in the form of *probability maps* generated using a further extension of the Visuset software system coupled with Google Earth[1]; the latter is so as to present the probability maps in the context of geographical locations. These two modules were incorporated into the framework. A drill down mechanism is also proposed so that users can focus their investigation of how information percolates across a selected subset of nodes in a given network. The conceptualisation and nature of both modules, and the associated evaluation, are described in detail in this chapter. The evaluation was again conducted using the GB cattle movement dataset that forms the central element of the Cattle Tracing System (CTS) in operation in England, Wales and Scotland. The CTS network was selected because: (i) it is the largest dataset considered in this thesis, and (ii) it was envisaged that the nature of its complex star form (as described in Chapter 3) would provide more interesting probability maps.

The rest of this chapter is organized as follows. In Section 6.1 some background on types of patterns that are required for use with the proposed prediction modeling is presented. Section 6.2 discusses the Percolation Matrix in detail. Then Section 6.3 describes the visualization module. In Section 6.4 the application of the “drill down” mechanism, that allows users to focus on a specific group of patterns based on their spatial attributes, is presented. Section 6.5 presents the results from the experimental analysis of the two modules that make up the prediction element of the PTMF. Finally, in Section 6.6 the chapter is concluded with a brief summary, some discussion and conclusions.

6.1 Background

The proposed prediction modeling mechanism is founded on the frequent patterns and trends generated using the TM-TFP algorithm in the Trend Identification module. As described in Chapter 4, the frequent pattern trends are identified from the analysis of a sequence of social network datasets. Recall that each frequent pattern trend is described in terms of its temporal occurrence counts (support values). This section comprises two sub-sections. Sub-section 6.1.1 defines the nature of the patterns that may be processed using the PM modules. Sub-section 6.1.2 presents an overview of the proposed PM process.

6.1.1 CTS Frequent Patterns and Trends

If we wish to model how information percolates across a network based on identified frequent patterns that exist in that network the patterns of interest must clearly include information about start and end locations and the nature of the traffic. From the Trend Identification module introduced in Chapter 4, a large number of frequent patterns and trends may be discovered from a given social network. Two types of attribute were

[1] `http://www.google.co.uk/intl/en_uk/earth/index.html`

considered:

1. Node Attributes (Location Attributes): Attributes associated with a node, in the case of the CTS application, examples include animal holding area IDs and animal holding area types.
2. Link Attributes (Movement Attributes): Attributes associated with links, in the case of the CTS application examples include breed, animal age and gender.

These two categories of attribute give rise to the concept of Location (Node) Patterns and Movement (Link) Patterns. Location patterns describe some aspect associated with locations. Movement patterns describe some aspect of movement. We can also identify Combination Patterns, patterns that comprise both location and movement attributes. For the purpose of the proposed prediction modeling we are interested in "traffic flow" between nodes, thus the type of patterns we are interested in are combination pattern that comprise location attributes associated with two different locations (nodes) and movement attributes concerned with the flow of activity between these locations. Therefore, for prediction purposes, the network needs to be conceptualised as comprising combination patterns of the form:

$$\{L_{fromLocation},\, M,\, L_{toLocation}\}$$

where $L_{fromLocation}$ is a specific value associated with a "from location" attribute, $L_{toLocation}$ is a specific value associated with a specific "to location" attribute, and M is some subset of the global set of movement attributes. In a complex star network the set of available values for the to and from location attributes are normally identical, in the case of a simple star network there is only one to location value and many from location values. The set M may consist of one or more attributes ($|M| \geq 1$). With respect to the work described in this thesis the $L_{fromLocation}$ and $L_{toLocation}$ attribute values were taken from the set of all possible values describing a set of possible location areas each identified by a unique number, and each defined in terms of a grid square delimited by an easting and northing coordinate system. An example is given in Figure 6.1 which features 25 grid square areas, $\{1, 2, 3, 4, 5, \ldots, 25\}$, each measuring 50 by 50km. Each grid square may hold zero, one or more cattle holding areas. The usage of an attribute such as the Location Area attribute was found to be desirable because: (i) little meaning (at least in the context of prediction mining) can be attached to predictions focused at the node level (a higher level of granularity is required) and (ii) it has the effect of reducing the overall number nodes by creating *super-nodes* (each describing a locality). Note also that the resulting network can be described in a tabular form. An exemplar of a combination pattern of the above form is {Sender Area = 12, Breed = Friesian, Number Cattle Animal $\leq$ 5, Receiver Area = 14}, a pattern which describes movements of cattle of breed Friesian in a quantity greater or equal to 5, from grid location 12 to grid location 14.

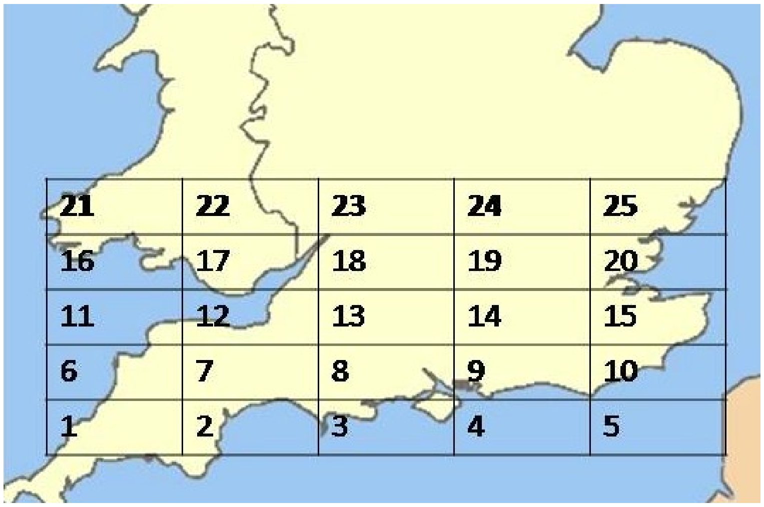


Figure 6.1: Simplified view of a map presenting 25 grid square locations

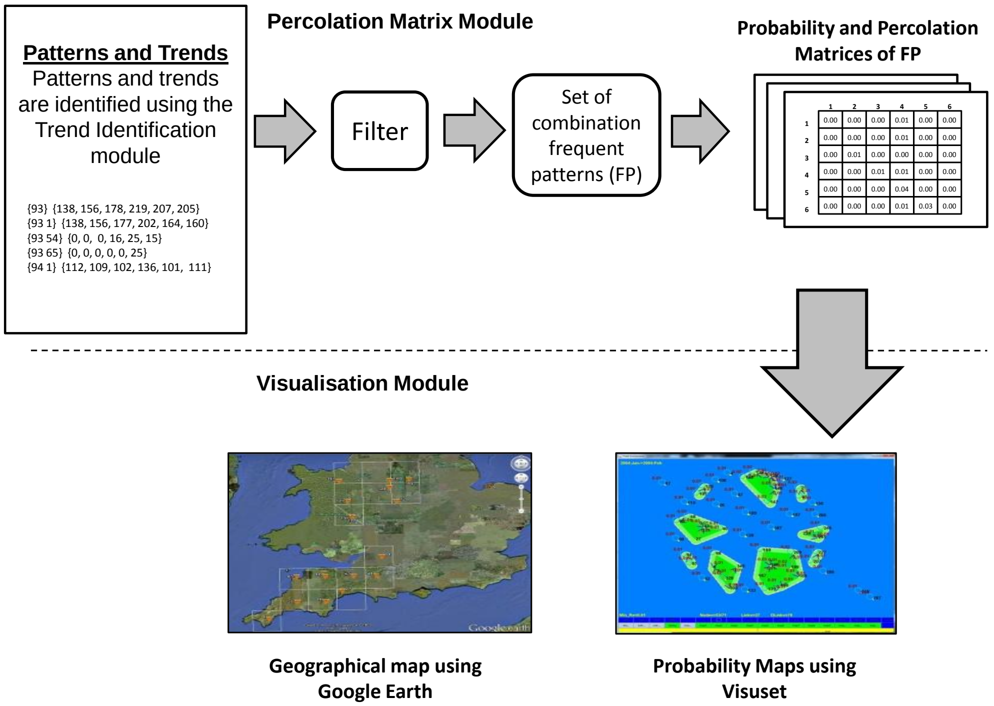
Each relevant combination pattern will have an associated trend line defined in terms of a set of support values. The support counts are indicators of the frequency of traffic between the two nodes. This in turn may be interpreted as a probability measure indicating the likely traffic flow between the two indicated locations. The M attributes may also be used to filter specific types of traffic of interest.

6.1.2 Prediction Modeling overview

Figure 6.2 gives a block diagram describing the proposed prediction modeling process. The process takes as input the identified frequent pattern trends. These trends are then filtered so that only the desired combination patterns remain as directed by the users' interests. A set of Percolation Matrices are then generated (this is described in the following section). A Percolation Matrix is formed by rows and columns representing the $L_{fromLocation}$ and $L_{toLocation}$ values, with the intersection of a row and a column holding the probability of traffic flow between the indicated nodes. The Percolation Matrix module produces a set of n percolation matrices (one per time stamp) indicating the probability of traffic flows from one node to another node. The final stage of the PM process is the Visualisation module. The Visualisation module can generate two types of map: (i) "probability maps" generated using a further extension of the Visuset software system, and (ii) geographical maps using Google Earth.

6.2 Percolation Matrix Module

In this section, the proposed Percolation Matrix Module is introduced. The module is intended to provide support for network analysis by indicating the probability of


Patterns and Trends
Patterns and trends are identified using the Trend Identification module
Percolation Matrix Module
Filter
Set of combination frequent patterns (FP)
Probability and Percolation Matrices of FP
Visualisation Module
Geographical map using Google Earth
Probability Maps using Visuset

{93} {138, 156, 178, 219, 207, 205}
{93 1} {138, 156, 177, 202, 164, 160}
{93 54} {0, 0, 0, 16, 25, 15}
{93 65} {0, 0, 0, 0, 0, 25}
{94 1} {112, 109, 102, 136, 101, 111}

	1	2	3	4	5	6
1	0.00	0.00	0.00	0.01	0.00	0.00
2	0.00	0.00	0.00	0.01	0.00	0.00
3	0.00	0.01	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	0.01	0.00	0.00
5	0.00	0.00	0.00	0.04	0.00	0.00
6	0.00	0.00	0.00	0.01	0.03	0.00

Figure 6.2: Block Diagram Indicating The Prediction Modeling (PM) Process

information or events traveling between nodes in a network. This information can then be used to determine the probability of traffic flow between three or more nodes. As already noted in the work described in this thesis, the probability is derived from the identified trend data.

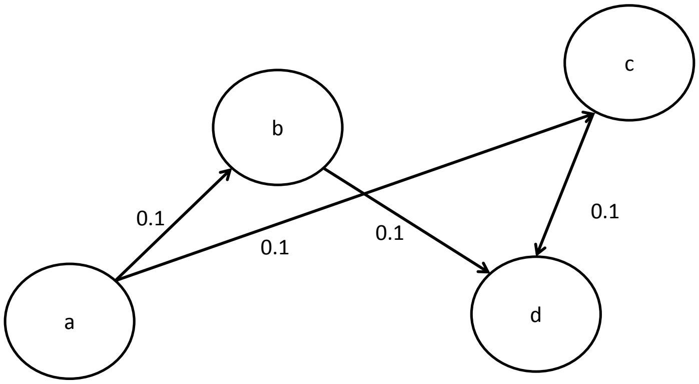


Figure 6.3: Conceptual Example of the Percolation of Information and Events in a network fragment

In Figure 6.3 an example of a snapshot of the nodes and links in a network fragment is presented. The figure shows a network of four nodes labeled $\{a, b, c, d\}$, connected by four links. The links are annotated with the probability of traffic flowing along this link at the given time stamp. This information can also be interpreted as the probability of a node being directly connected to another node. Similarly, combinations of such probabilities can indicate indirect connections between nodes. Thus, referring to Figure 6.3, the probability that node a is connected to node b is given as 0.1. Thus there is a possibility of 0.1 that some piece of information or event occurring at node a will travel to node b. Similarly the probability that an event occurring at node a will be transmitted to node d is $0.1 \times 0.1 = 0.01$.

This section consists of two sub-sections describing the Percolation Matrix module. In Sub-section 6.2.1, the process of filtering frequent pattern trends is described, then Sub-section 6.2.2 explains the process of transcribing the probabilities associated with a set of combination patterns to form the percolation matrix.

6.2.1 Filtering The Frequent Patterns

The Percolation Matrix module starts with the process of filtering frequent patterns and trends to be used in the PM. As mentioned in Sub-section 6.1.1, the frequent patterns of interest are combination patterns of the form: $\{L_{fromLocation},\ M,\ L_{toLocation}\}$. The process of generating the set of combination patterns of interest (FP) is dependent on the interest of the domain user. Typically the selection is based on some constraints to be applied so as to filter the global set of movement patterns. For example if $M = \{m_1, m_2, m_3\}$ and the set of location values is $\{a, b, c, d\}$ then the set of combination patterns might be:

$$\begin{aligned} FP = \{&\{a, m_1, m_2, m_3, b\}, \\ &\{a, m_1, m_2, m_3, c\}, \\ &\{b, m_1, m_2, m_3, d\}, \\ &\{c, m_1, m_2, m_3, d\}\} \end{aligned}$$

The set FP and the associated trends are then used as input to the Percolation Matrix module.

6.2.2 Probability and Percolation Matrices

The second part of the Percolation Matrix module comprises a two stage processes: (i) determine the probability of link traffic in the set FP, and (ii) construct the desired n percolation matrices for FP. As mentioned earlier, the trends for each fp_i in FP are used to compute the probability of link traffic. Then given a n time stamp trend, n percolation matrices will be generated. A percolation matrix consists of a $N \times N$ elements, where $N = \{n_1, n_2, \ldots, n_n\}$ is the number of possible location pattern values. The magnitude of N is dependent on the number of distinct $L_{fromLocation}$ and $L_{toLocation}$

contained in FP. The intersection of a row and column in the matrix indicates the probability value of associated link traffic.

Algorithm 6.1: The Probability and Percolation Matrix

```
  input  : $FP$ = Set of Frequent combination patterns, set of Trends
  output: $n$ Percolation Matrices generated from $FP$

1 for $\forall fp \in FP$ do
2     Extract probability ($p$) of each $fp$ from its associated trend;
3 end
4 for $k \leftarrow 1$ to $\mid Trends \mid$ do
5     Construct a matrix of size $N \times N$);
6     for $i \leftarrow 1$ to $\mid FP \mid$ do
7         Insert $p_i$ into the matrix$_k$ at the appropriate location;
8     end
9 end
```

Algorithm 6.1 describes the process of extracting the probability of information movement and building the percolation matrix to facilitate the desired Prediction Modeling. The algorithm first extracts the probability of each pattern fp in FP (Line 2). The support values associated with each time stamp n defines the probability of traffic flowing between nodes. Thus all support values for the selected frequent patterns are converted into a probability value (p). Therefore, given a specific frequent pattern fp_i, conforming to some types of combination pattern, p_i for fp_i is defined as:

$$p_i = \frac{support(fp_i)}{\sum_{fp_i}} \tag{6.1}$$

Thus, $p_1 + p_2 + \ldots + p_n = 1$.

Once the probability for all fp has been extracted, the algorithm constructs the percolation matrix (Line 3). As already noted, the size of the matrix is dependent on the number of available values for $L_{fromLocation}$ and $L_{toLocation}$. Then the probabilities of traffic associated with all fp are inserted into the matrix. The process repeats until all n percolation matrices are constructed. Table 6.1 shows an example of the output of Algorithm 6.1 with respect to the network fragment present in Figure 6.3. These percolation matrices are then used as the input to the Visualisation module described in the next section.

From/To	a	b	c	d
a	0	0.1	0.1	0
b	0	0	0	0.1
c	0	0	0	0.1
d	0	0	0	0

Table 6.1: An example of a Percolation Matrix using the network fragment given in Figure 6.3

6.3 Visualization Module

The Visualisation module includes two types of visualisation tool, both directed at illustrating the content of the percolation matrix.

1. **The Visuset Prediction Map Tool**: A customized version of Visuset that illustrates the way traffic is likely to flow across a network (Sub-section 6.3.1).

2. **The Geographical Map Tool**: A presentation tool that produces a Google Earth "overlay" (Sub-section 6.3.2).

6.3.1 The Visuset Prediction Map Tool

The aim of the Visuset tool is to demonstrate, in a clear and straight forward manner, how information travels across a given network. As noted previously, Visuset is a "2-D drawing area" visualisation software system that displays nodes and node communities using a Spring Model [10]. The customised Visuset system provides an interpretation of a probability matrix in the form of a *probability map* that illustrates a given node and link structure as shown in Figure 6.3. The maps highlight which nodes are connected directly (and, by extension, indirectly) to other nodes using "weighted" links. The weightings are determined from the probabilities contained in the generated percolation matrices. The configuration of Visuset used for the purpose of prediction modeling is similar to the configuration of Visuset used in Chapter 4. However, in this case, the numbers of patterns in each node are ignored as the significant information to be displayed are location nodes, traffic links and probability values from the percolation matrices.

As mentioned earlier, the extension of Visuset includes a mechanism to identify communities of nodes in the network. However, using the percolation matrix it is straightforward to identify nodes that are connected together as this information can be extracted directly from the matrix. Thus, the probability maps illustrate groups of nodes that are connected together which, in the same manner as described previously in chapter 4, are depicted as "islands". From the generated maps, the following can be identified:

1. Paths describing how information or events may travel between nodes (in one "step"). In the proposed PM, a *one step percolation* is defined as a direct link between a pair of nodes, a and b.

2. Probability values describing the likelihood that information may travel between a particular pair of nodes. The probability values can also be used to calculate the probability of information flows encompassing two or more steps. A *two steps percolation* refers to movement between two pairs of connected nodes. For

example, in Figure 6.3, movement patterns may percolate from node a to node d through node c. We may also be able to identify three and four steps percolations (n-step percolations). These types of percolation are collectively described as "complex" connections.

3. Communities of nodes that are connected together.

6.3.2 The Geographical Map tool

The Geographical Map tool is directed at providing a Google Earth overlay that relates the prediction map (introduced above) to actual geographic locations. Google Earth is sometimes referred to as a "virtual globe" that allows users to "fly" and explore 3-D images of the surface of the earth. Previously known as Earth Viewer 3D, what we now know as Google Earth was created by Keyhole Inc. Google acquired Google Earth in 2004. There are a number of reports where researchers have utilised Google Earth. For example Honjo *et al.* [54] proposed a landscape visualisation system using Google Earth to act as a practical and low cost landscape simulation tool. Another example, Multigesture.net, introduced the concept of Earth Friends, a free Facebook application for locating "friends" using Google Earth [2].

In the context of the work described in this thesis, Google Earth is used to highlight geographical locations described by the value set for $L_{fromLocation}$ and $L_{toLocation}$. An Earth imagery (map) Google Earth overlay can be implemented using the Keyhole Markup Language (KML). KML is the file format that is used to layout geographic data in the "earth browser" used by both Google Earth and Google Maps. KML uses an XML style notation which uses a tag-based structure with nested elements and attributes. The elements in KML allow landmarks, grid lines, labels and so on to be placed over the earth imagery provided by Google Earth. More details concerning KML can be found in [127].

With respect to the work described in this thesis the process of customising the KML file is very specific to the nature of the node patterns discovered within a given social network. This is because the KML source will require the terrestrial coordinates (latitude and longitude) of the $L_{fromLocation}$ and $L_{toLocation}$ values. Thus, each generated probability map will have an individual customised KML file associated with it.

6.4 Drilling Down

It was considered useful to also allow users to "drill down" in a specific geographic area so that a more detaile view can be provided. This was achieved by providing a facility to allow users to dynamically change the location area grid size (set at 50km by default). The effect is achieved by introducing a sub-division of a location area into sub-areas.

To illustrate the drill down method, some node patterns generated from the CTS network will be used. Thus if we select the location areas {127, 128, 148, 149} we can drill-down to produce 16 new sub-areas (nodes). If we consider location area (node ID) 127 this may be interpreted as representing four sub-areas which we might name: {127(SW), 127(SE), 127(NE), 127(NW)}. To identify the trends in these new areas the process describing in Chapter 4 will need to be repeated. These identified patterns are then the input for a repeat of the PM process. It is also possible to repeat the drill down process to an even smaller grid size for further investigation of a specific location area.

6.5 Experimental Analysis of The Prediction Modeling

This section describes the experimental analysis of the two PM modules described above (the Percolation Matrix module and the Visualisation module). The CTS network dataset was used for the evaluation. Several combination patterns were selected as input to the PM tasks. The combination patterns are filtered using 3 different constraints. The evaluation started with the generation of appropriate percolation matrices, the results were then used by the Visualisation module. Most of the experimental analysis was undertaken using 50km location areas each defined by a unique numeric identifier, the complete set of numeric identifiers then described the set of values from which the values for $L_{fromLocation}$ and $L_{toLocation}$ were drawn. The identifiers we also used as the node identifiers. To illustrate the "drill down" facility each 50km grid square was subdivided into 25km sub-grid squares (to allow for observation of cattle movements in more detail). Note that only selected examples of the generated percolation matrices and visualizations can be presented here.

Sub-section 6.5.1 describes the nature of the selected CTS combination patterns used for the evaluation. Then Sub-section 6.5.2 reports on the analysis of the percolation matrix generation process, whilst Sub-section 6.5.3 presents the evaluation of the visualisation module. Lastly, Sub-section 6.5.4 considers the analyses of the drill-down facility using a specific group of CTS node patterns.

6.5.1 Frequent Patterns Selection

Recall that the prediction modeling only operates using a specific set of combination patterns, FP, comprised of to and from location attribute values and a movement pattern of some kind. With respect to the evaluation described here using the CTS network the following "permitted" combinations were adopted:

1. **Type 1** $= \{Sender\ Area,\ Animal\ Age = allAnimal\ Age\ sub\ patterns,$ $Breed\ = all\ Breed\ sub\ patterns,$ $Number\ Animals\ Moved = all\ Number\ Animal\ Moved\ sub\ patterns,$ $Receiver\ Area\}.$
2. **Type 2** $= \{Sender\ Area, Number\ Animals\ Moved\ \leq 5, Receiver\ Area\}.$
3. **Type 3** $= \{Sender\ Area, Breed\ = Luing, Number\ Animals\ Moved\ \leq 5,$ $Receiver\ Area\}.$

where $\{Sender\ Area$ and $Receiver\ Area\}$ are location attributes and $\{Animal\ Age$, $Breed$, $Number\ Animals\ Moved\}$ are movement attributes associated with a movement pattern. It is of course possible to identify alternative (application dependent) combination patterns. As already noted the significance of combination patterns is that they define movement between locations. The probability of a movement occurring is then defined by the support counts for each pattern.

6.5.2 Percolation Matrix

This sub-section presents an analysis of the use of percolation matrices. With respect to the CTS network used for the evaluation the number of time stamps considered was 12 (months) thus, the Percolation Matrix module produced 12 monthly percolation matrices. A number of examples of the generated percolation matrices, using the above Type 2 combination patterns, are considered in this sub-section. Figure 6.4 provides a comparison of run times recorded when generating the percolation matrices.

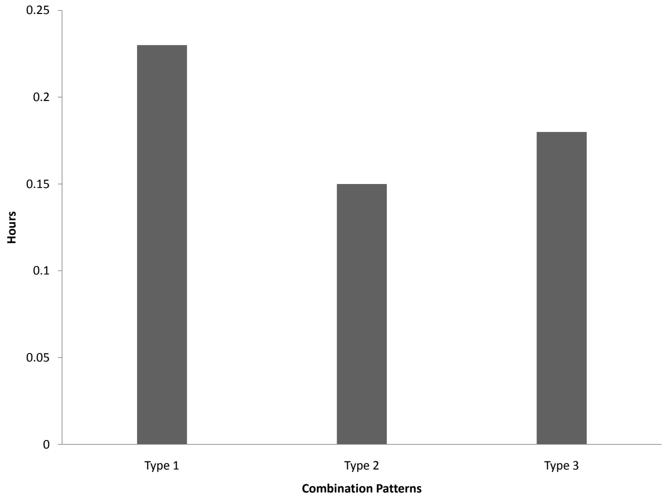


Figure 6.4: The Percolation Matrix run time values (seconds) comparison

The Percolation Matrix module commences by extracting the probability values from the trends for each fp_i in FP. Table 6.2 and 6.3 list some example probability (support) values for CTS Type 2 combination patterns, between January 2003 and May 2003, associated with particular to and from location areas. The location areas are identified by a subset of the available location area (node) IDs: {127, 128, 147, 148, 149, 168, 169, 187}. Table 6.4 provides the easting and northing details of the location area/node IDs. From Table 6.2 and 6.3 it can be noted that most of the cattle movements happen within the same location area grid square, however there are some cattle movements that cross to adjacent grid squares, for example from node 127 to 128. Thus, from Table 6.2, it can be deduced that if an event (for example the detection of an animal disease) occurs at location area 127 it will be passed to the adjacent location area of node 128 with a probability of 0.05, because of the support value associated with the connecting link between the two nodes. This form of percolation is referred to as a *one step percolation*. Likewise, in Table 6.3, if an event occurs in location area 147 it is likely that it will be transmitted within the same location area with a probability of 0.03.

Sender	Receiver	Jan	Feb	Mar	Apr	May
127	128	0.05	0.06	0.05	0.05	0.05
147	147	0.09	0.08	0.09	0.08	0.07
148	148	0.07	0.07	0.07	0.07	0.07
149	149	0.03	0.02	0.02	0	0
168	168	0.05	0.05	0.05	0.04	0.04
169	169	0.04	0.04	0.05	0.04	0.04
187	187	0.03	0.03	0.03	0.03	0.02

Table 6.2: Sample of 2003 CTS Type 2 combination pattern Monthly Probabilities

Sender	Receiver	Jan	Feb	Mar	Apr	May
127	128	0.02	0.02	0.02	0.02	0
147	147	0.03	0.03	0.03	0.04	0.04
148	148	0.06	0.06	0.06	0.06	0.07
149	149	0.04	0.04	0.05	0.03	0.04
168	168	0.06	0.06	0.06	0.05	0.06
169	169	0.03	0.03	0.03	0.02	0.03
187	187	0.02	0	0	0	0

Table 6.3: Sample of 2004 CTS Type 2 combination pattern Monthly Probabilities

The next stage is to generate the n percolating matrices, an example fragment of the percolation matrix generated using the CTS dataset is shown in Table 6.5. From the matrix shown in Table 6.5, it can be observed that if an event occurs at location area 127 it is likely to infect all holdings within this location area with a probability of 0.04, and infect location area 128 with a probability 0.02. Another example is that if an event occurs at location area 128 it will infect other holdings in the same location

Sender/Receiver Areas	easting and northing (in meters)
127	easting (301000-350000) and northing (301000-350000)
128	easting (351000-400000) and northing (301000-350000)
147	easting (301000-350000) and northing (351000-400000)
148	easting (351000-400000) and northing (351000-400000)
149	easting (401000-450000) and northing (351000-400000)
168	easting (351000-400000) and northing (401000-450000)
169	easting (401000-450000) and northing (401000-450000)
187	easting (301000-350000) and northing (451000-500000)

Table 6.4: Definition of Example Location Area Grid IDs in Terms of Eastings and Northings

area with a probability of 0.05, and infect location area 148 with a probability 0.02. Since all these predicted events will occur as a result of a single link between two nodes these are all referred to as *one step percolations*. In addition, it is possible to calculate the likelihood that an event occurring at (say) location area 127 will percolate to (say) location area 148. In this case the probability will be 0.0004 (the probability of node 127 infecting node 128 multiplied by the probability of node 128 infecting node 148, thus $0.02 \times 0.02 = 0.0004$) this is thus a *two steps percolation*.

From/To	127	128	147	148	149	168	169	187
127	0.04	0.02	0	0	0	0	0	0
128	0	0.05	0	0.02	0	0	0	0
147	0	0	0.03	0	0	0	0	0
148	0	0	0	0.06	0	0	0	0
149	0	0	0	0	0.04	0	0	0
168	0	0	0	0	0	0.06	0	0
169	0	0	0	0	0	0	0.03	0
187	0	0	0	0	0	0	0	0.02

Table 6.5: January 2004 Type 2 Percolation Matrix indicating the probability of an event "percolating" from one location area to another in n step

Given the above it can be seen how the proposed Percolation Matrix module can be successfully employed to identify how events may percolate across a network. This understanding can be further enhanced using the proposed Visualization module. The evaluation of this module is presented in the next section.

6.5.3 Visualisation of Prediction Modeling

This section presents the evaluation of the Visualisation module. The evaluation was conducted by considering different types of probability map sequences: (i) monthly probability map sequences and (ii) yearly probability map sequences. The first is considered in Sub-section 6.5.3.1 and the second in Sub-section 6.5.3.2. The selected results presented in Sub-sections 6.5.3.1 are from between January and May 2003 for CTS Type 1, Type 2 and Type 3 combination patterns. Whereas in Sub-section 6.5.3.2, the results presented are for January 2003, 2004, 2005 and 2006. The rest of the probability maps for CTS Type 1 and Type 2 between February and December 2003, 2004, 2005 and 2006 can be found in Appendix A to H.

6.5.3.1 Evaluation of Monthly Prediction Modeling

So as to evaluate the analytical support provided by the monthly probability maps all three identified types of combination pattern were considered in turn.

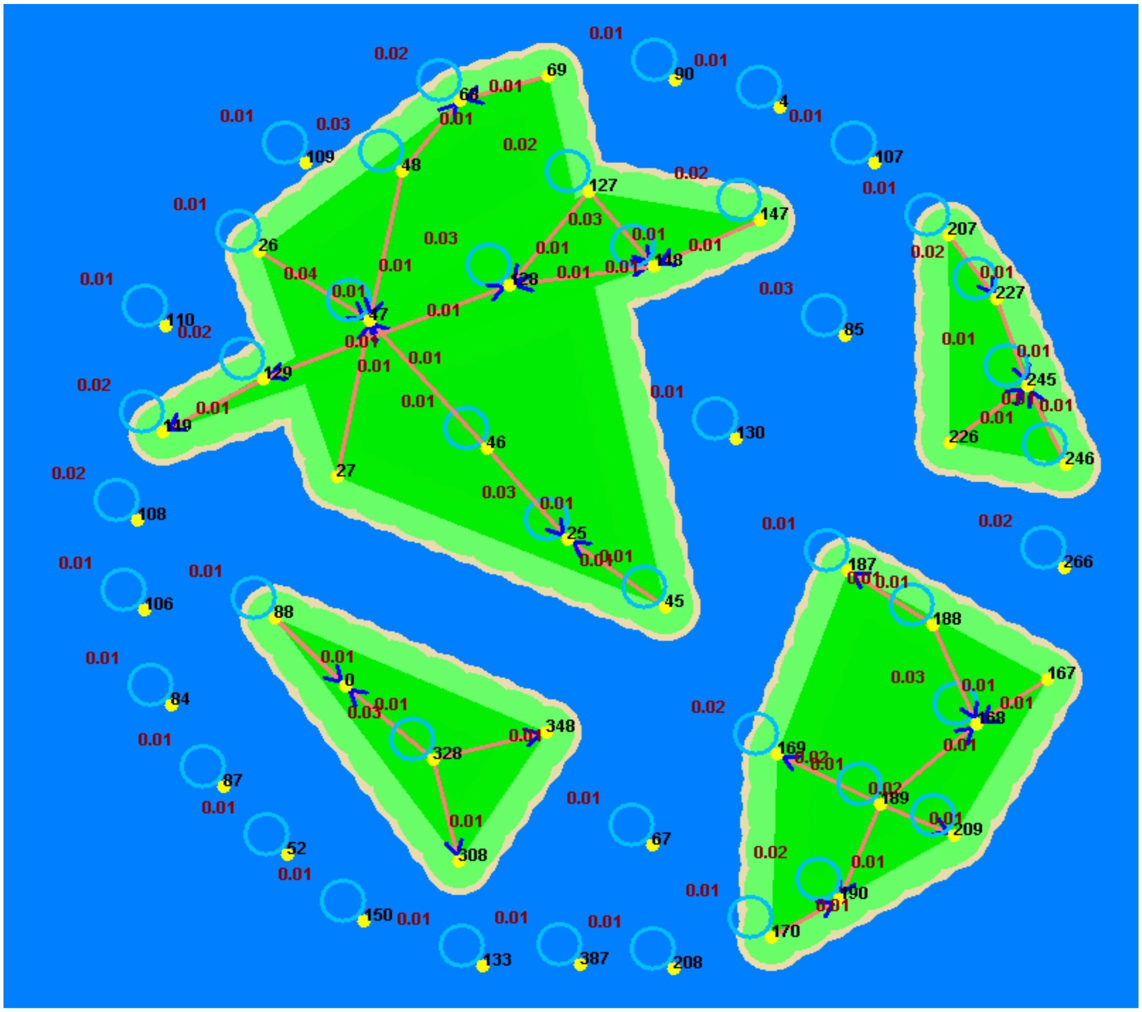


Figure 6.5: January 2003 Type 1 Combination Patterns Probability Map

Type 1 CTS combination pattern ($\{Sender\ Area, Animal\ Age = all\ Animal\ Age\ sub\ patterns, Breed\ Type = all\ Breed\ Type\ sub\ patterns, Number\ Animal\ Moved = all\ Number\ Animal\ Moved\ sub\ patterns, Receiver\ Area\}$) were considered first. Figures 6.5, 6.6, 6.7, 6.8 and 6.9 show the probability maps generated for a sequence of five time stamps covering the period from January to May

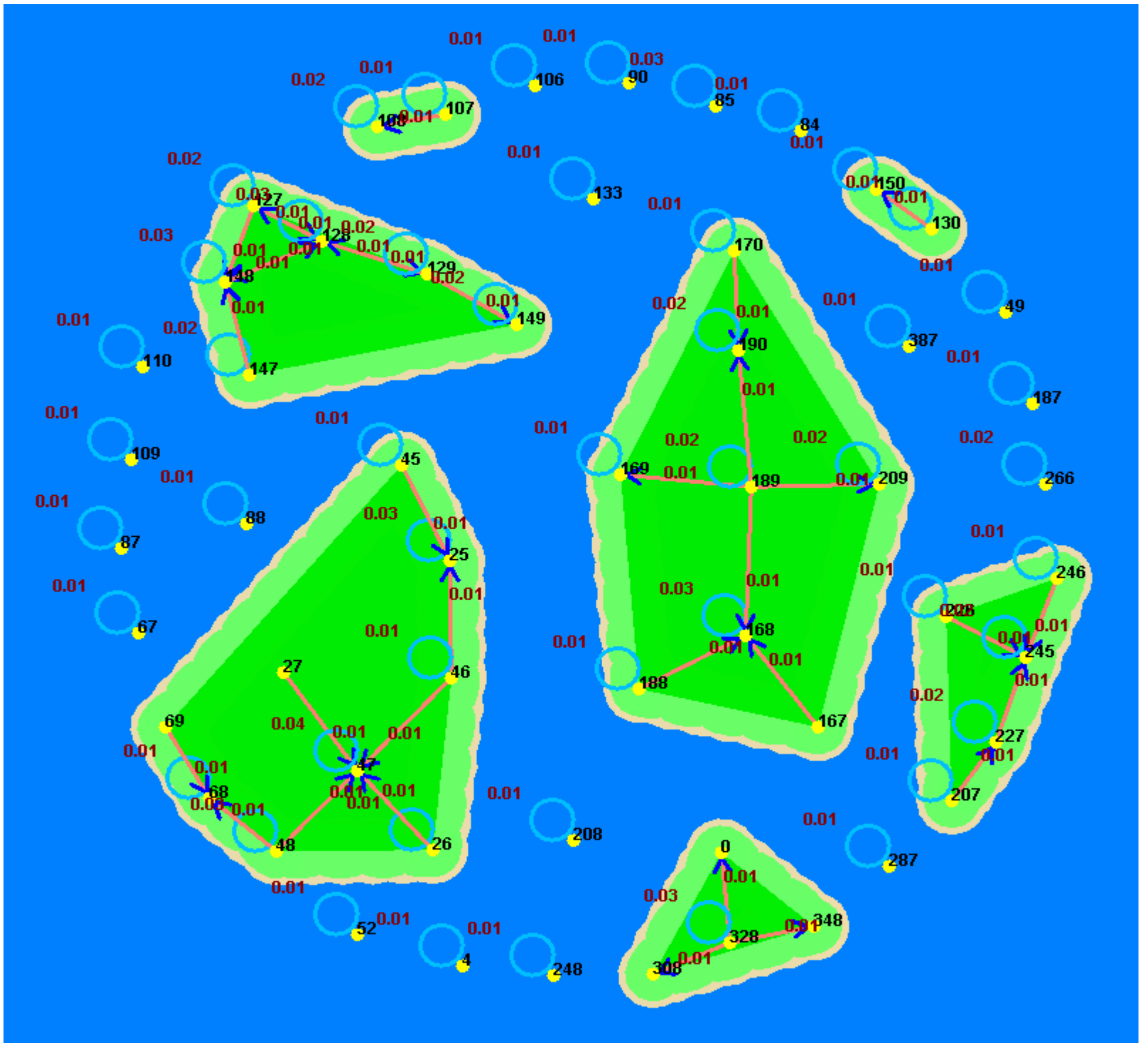


Figure 6.6: February 2003 Type 1 Combination Patterns Probability Map

2003. Note that, in all of the probability maps $L_{fromLocation}$ and $L_{toLocation}$ values are used as the node labels. The links describe Type 1 movement patterns. From the maps it can be noted that most nodes in the maps have a probability of 0.01 that movement will occur within the same node. The maps include a number of “islands” of nodes that are connected together. The majority of these islands, in all the probability maps, comprise the same node patterns. Nevertheless there are a few new nodes that appeared or disappeared as the sequence progresses. In some case islands split into new smaller islands in a following month in the sequence. For example, the island comprised of location areas (nodes) $\{207, 226, 227, 245, 246\}$ in the maps for January, February and April 2003 was divided into two in March and May 2003. From the maps it can also be seen that there is a possibility that an event might originate from within several nodes that link to some receiver nodes, as in the case of location area 47 in Figure 6.5.

There were also indirect connections between nodes, such as between nodes 207 and

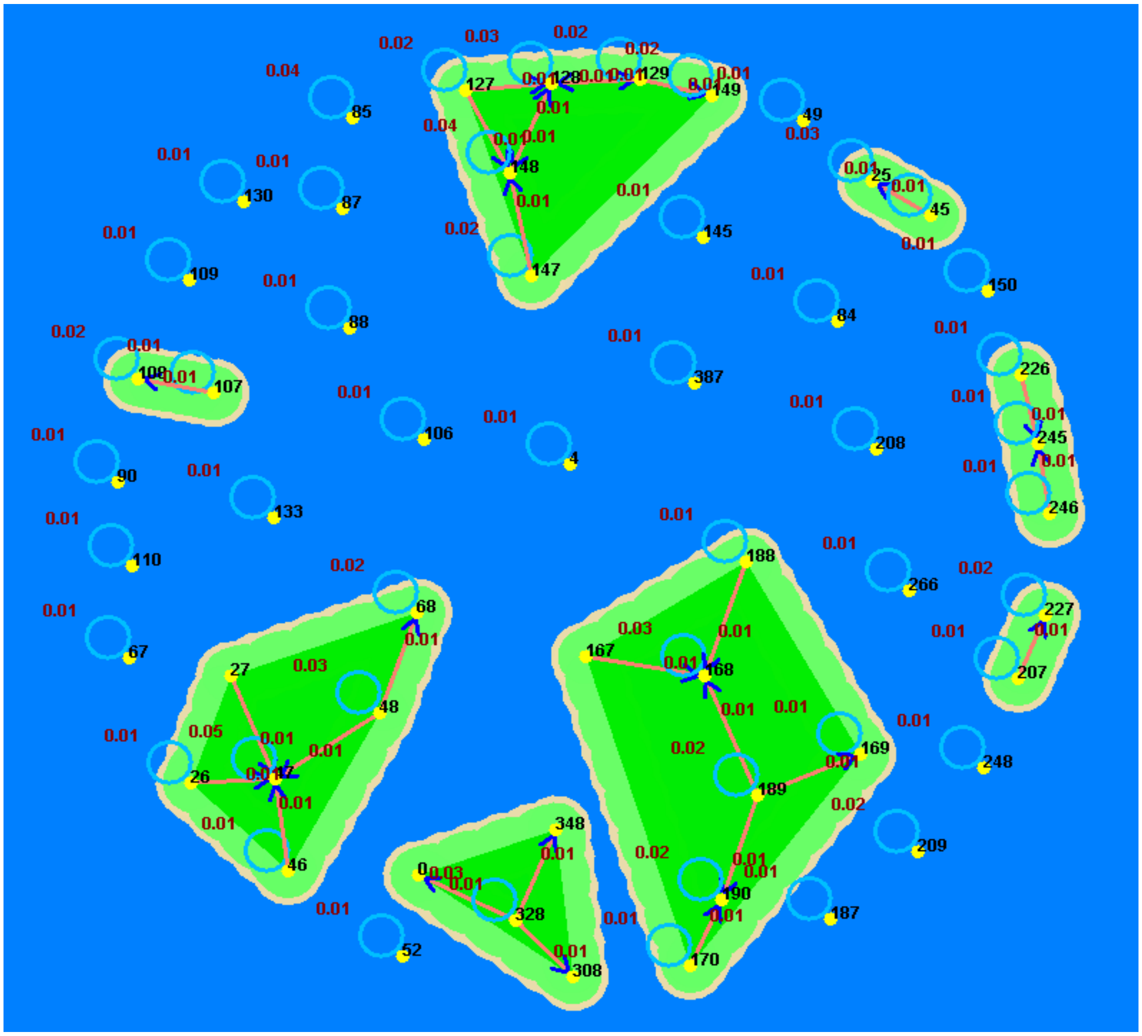


Figure 6.7: March 2003 Type 3 Combination Patterns Probability Map

245 in the January, February and April 2003 maps. Thus, considering Figure 6.5, if an event happens in location area 245, it could originate either from location area 227 in a *one step percolation* (with probability of 0.01) or from location area 207 in *two steps percolation*, with probability of 0.0001 ($0.01 \times 0.01 = 0.0001$). In terms of the identified islands if there is an event, for example an animal infection, it can be predicted that it is likely to spread within the detected islands of areas, but less likely to spread outside the islands.

The percolation matrices for CTS Type 2 combination patterns ($\{Sender\ Area, Number\ Animal\ Moved\ \leq 5, Receiver\ Area\}$) were also used to generate probability maps. Figures 6.10, 6.11, 6.12, 6.13 and 6.14 show the probability maps for between January and May 2003. Again, most of the movement patterns travel within the same location areas as the majority of nodes have self-links. In addition, in all the maps, only a few links existed between the adjacent areas. Inspection of all maps indicates that there is consistent link traffic from the location

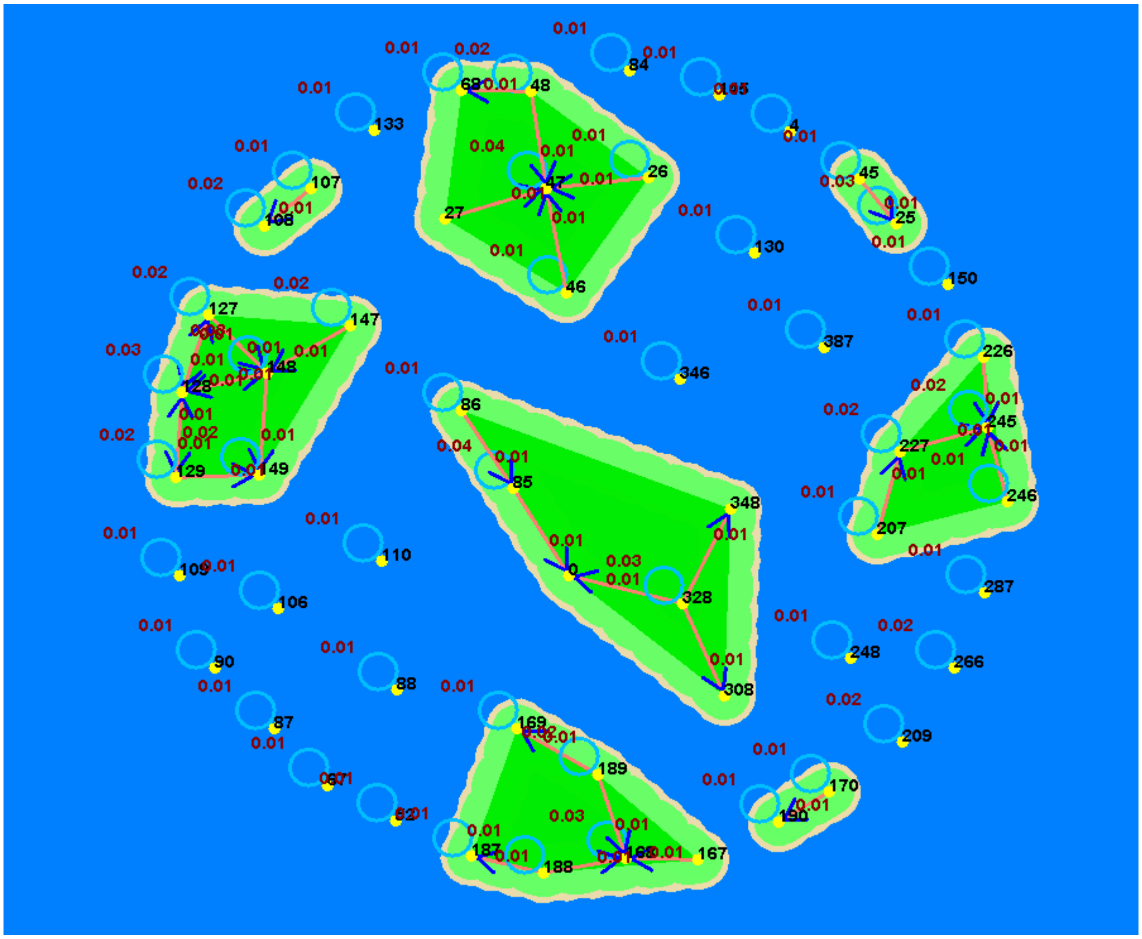


Figure 6.8: April 2003 Type 1 Combination Patterns Probability Map

area (node) 127 to the location area (node) 128, with probabilities of between 0.05 and 0.06 respectively. Figures 6.10 and 6.11 also show link traffic from location area 48 to location area 47 with a probability of 0.02. From Figure 6.14 it can also be deduced that cattle movements may occur within three location areas. There is a possibility that if an event, such as disease spread, happens in location area 148, that it might originate from location area 127 or 128. Even though location area 127 is not connected directly to 148, but it is connected in terms of a *two steps percolation* with a probability of 0.001.

Further evaluation was conducted using Type 3 combination patterns ($\{Sender\ Area, Breed\ Type = Luing, Number\ Animal\ Moved\ \leq 5, Receiver\ Area\}$). The Type 3 combination pattern had more criteria with which to filter the CTS frequent patterns, thus its usage resulted a smaller number of frequent patterns than in the case of Type 1 and Type 2 combination patterns. Figures 6.15, 6.16, 6.17 and 6.18 showed the probability maps generated for the period from January to April 2003. Inspection of the maps indicates that in this case all the cattle movements happen within the same nodes (location areas). Therefore, it is easy to identify which location areas

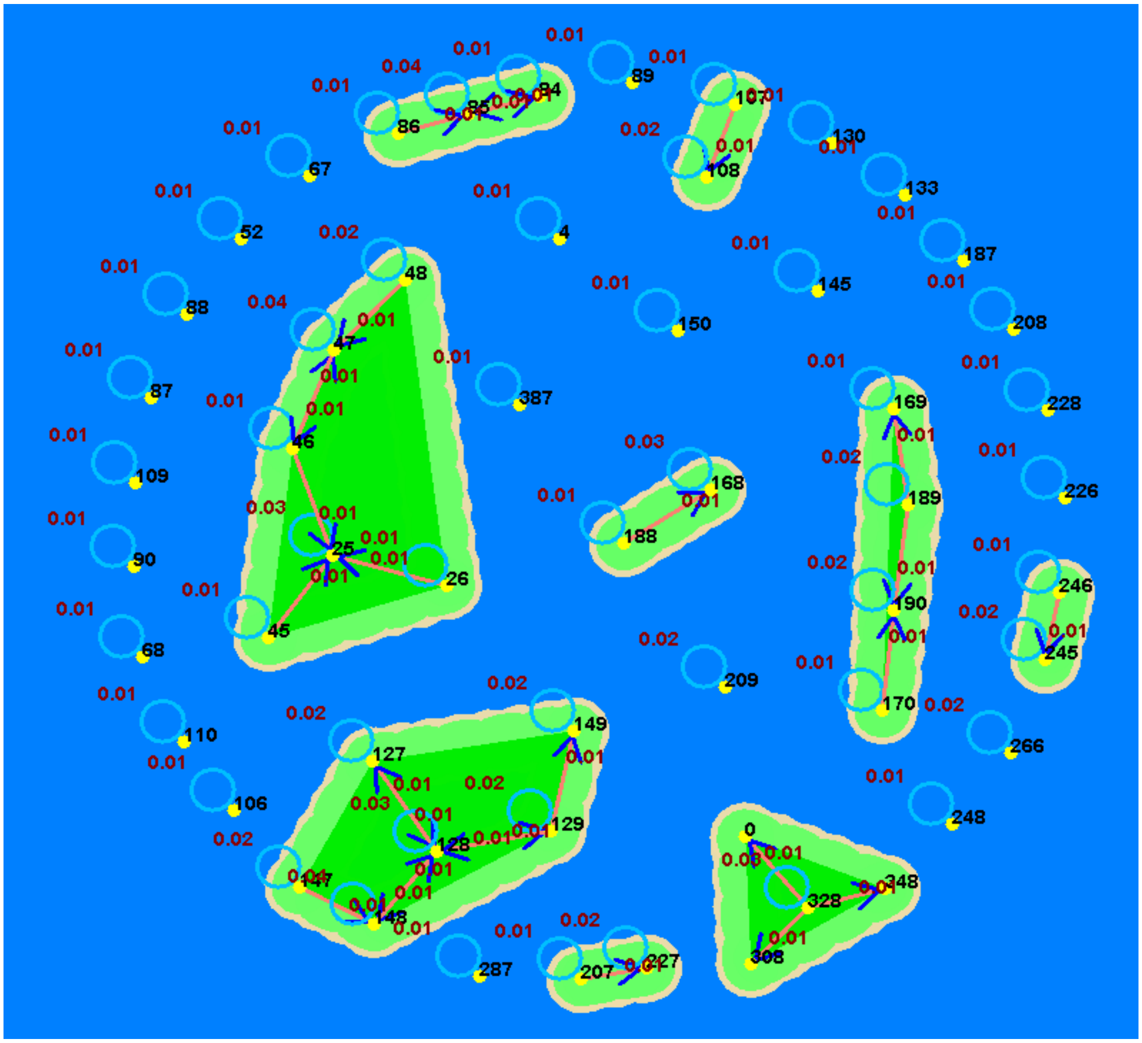


Figure 6.9: May 2003 Type 1 Combination Patterns Probability Map

have cattle movements with *Breed* = *Luing* (the Luing is a relatively rare beef cattle bread, it is very hardy and usually found in upland areas).

As mentioned above a Google Earth tool was included in the framework to relate areas to geographical locations. Most of the example probability maps presented above have similar node patterns, thus for evaluation of the Google Earth tool only the CTS Type 1 combination patterns between January and February 2003 are considered here. To produce the desired map the identified probability map needed to be first translated into a KML file, the result is as shown Figure 6.19. From the figure the relevant location areas in relation to the overall geography of GB can be clearly identified, as can the likely traffic flows between areas (indicated by red edges in the figure). The figure also indicate the areas within GB where significant cattle farming activities are concentrated. Discussion with domain experts has highlighted the usefulness of this display.

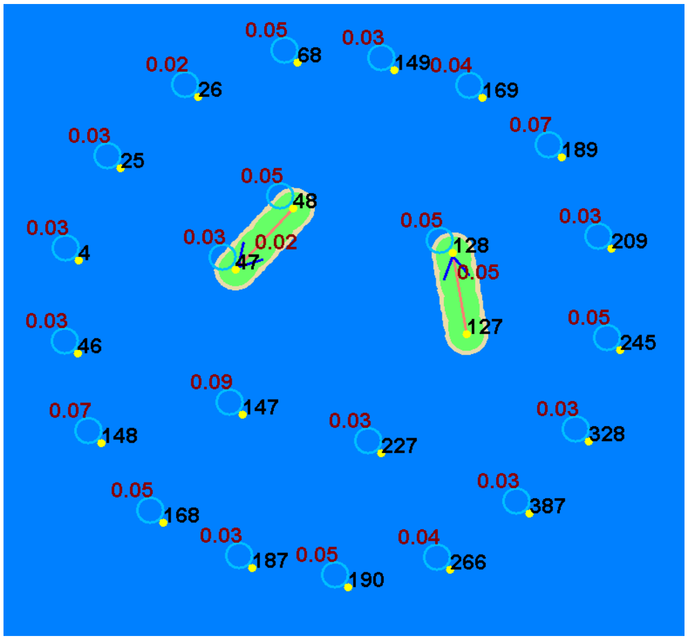


Figure 6.10: January 2003 Type 2 Combination Patterns Probability Map

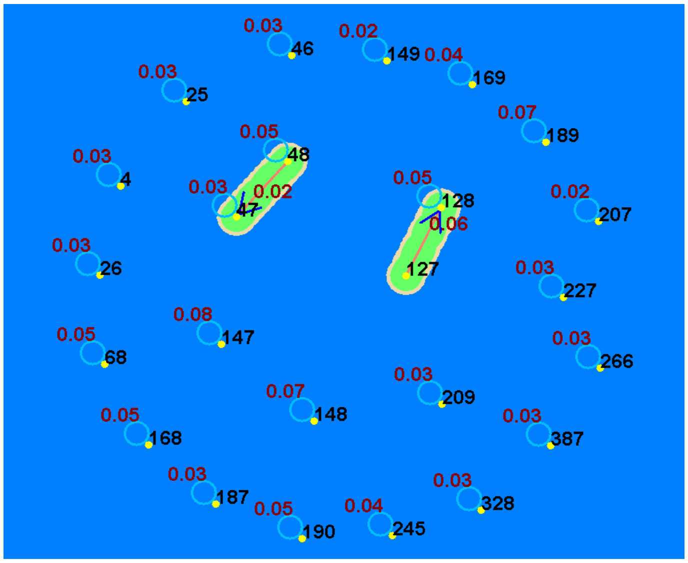


Figure 6.11: February 2003 Type 2 Combination Patterns Probability Map

6.5.3.2 Evaluation of Yearly Prediction Modeling

The Percolation Matrix module produces a sequence of percolation matrices which can be visualized in the form of prediction maps. The mechanism can also be applied to address “longitudinal” studies. Thus given a sequence of episodes (recall that we divide our time stamps into episodes) we can compare time stamp$_i$ in episode j with time stamp$_i$ in episode $j + 1$, and so on. The evaluation of longitudinal studies of the form reported in this chapter is again focused on the CTS data. The reported comparisons were made using Type 1 and Type 2 combination patterns. Figure 6.5,

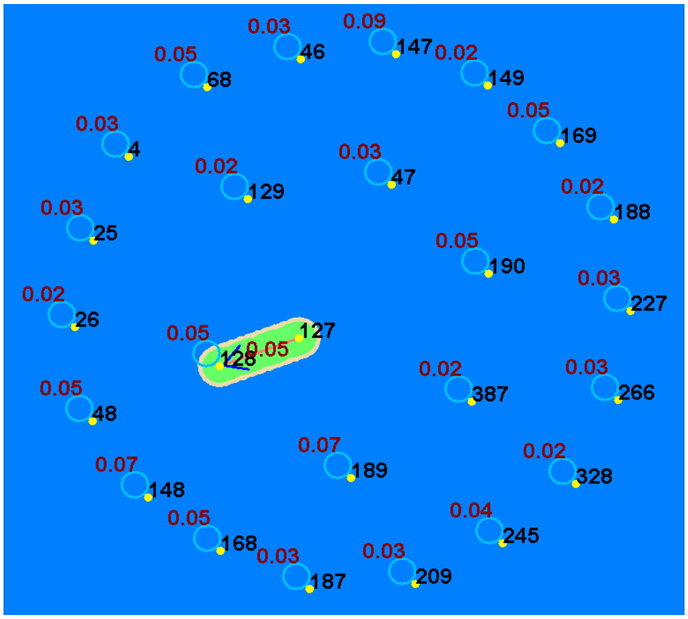


Figure 6.12: March 2003 Type 2 Combination Patterns Probability Map

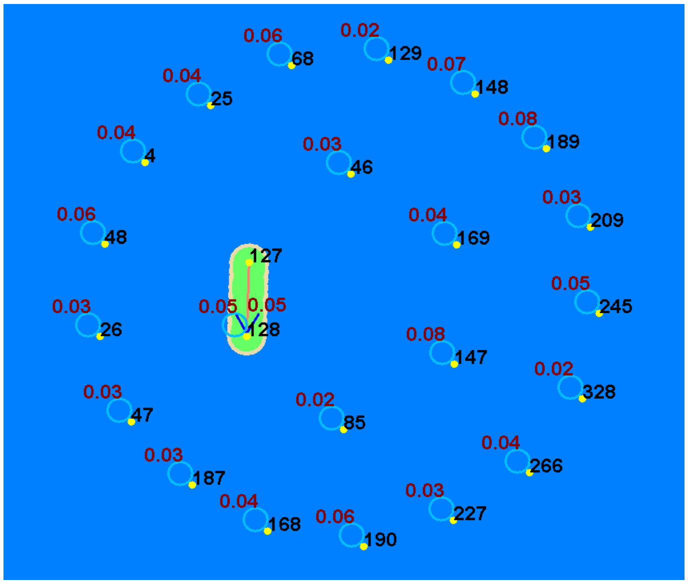


Figure 6.13: April 2003 Type 2 Combination Patterns Probability Map

6.20, 6.21 and 6.22 show the cattle movements for the month of January with respect to four consecutive episodes (2003, 2004, 2005, 2006) using the Type 1 combination patterns ($\{Sender\ Area, Animal\ Age\ =\ all Animal\ Age\ sub\ patterns, Breed\ =\ all\ Breed\ Type\ sub\ patterns, Number\ Animal\ Moved\ =\ all\ Number\ Animal\ Moved\ sub\ patterns, Receiver\ Area\}$). Again several islands can be observed in the maps to show how nodes in areas are connected directly (one step) and also indirectly (two steps). However, the yearly maps show slightly different types of islands, in terms of the number of areas and also the direction of links between them, demonstrating how the islands change over a number of years.

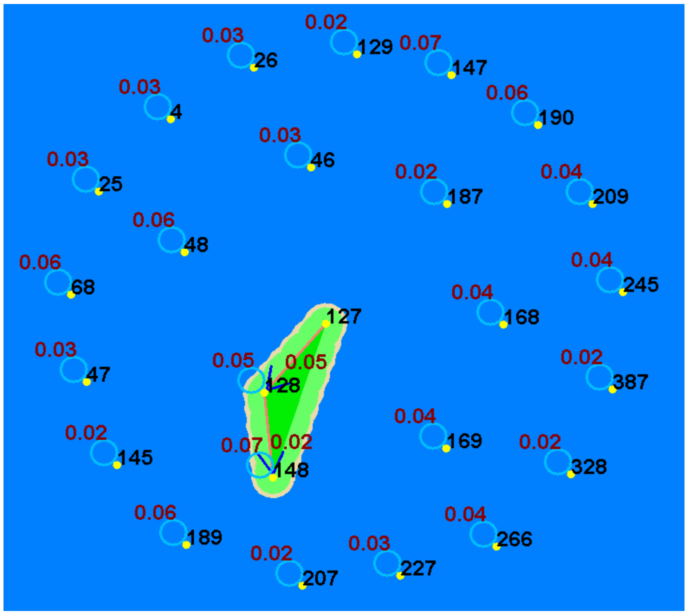
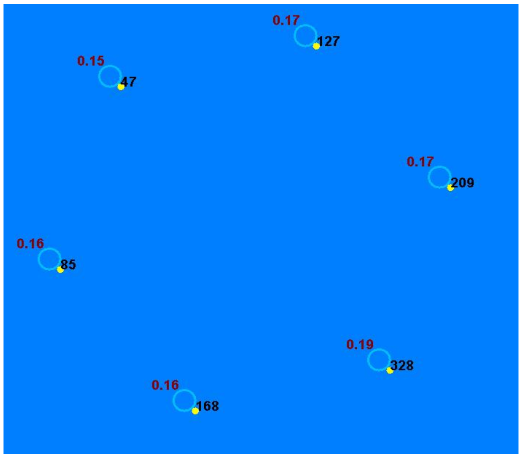


Figure 6.14: May 2003 Type 2 Combination Patterns Probability Map



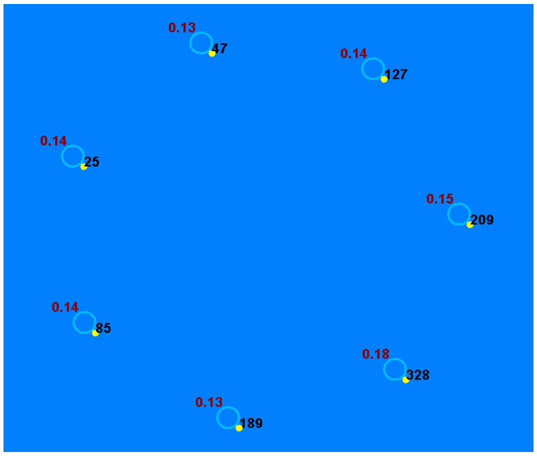


Figure 6.15: January 2003 Type 3 Combination Patterns Probability Map

Figure 6.16: February 2003 Type 3 Combination Patterns Probability Map

Inspection of Figure 6.20 indicates that in January 2004 there are more "complex" connections between nodes. For example events within location area 127 could reach location area 149 following two separate routes (via node 148, or via nodes 128 and 129), the first with a probability of 0.0002, the second with a probability of 0.000006.

Similarly, Figures 6.10, 6.23, 6.24 and 6.25 show yearly comparisons using Type 2 combination patterns {Sender Area, Number Animal Moved $\leq$ 5, Receiver Area} for the month of January. Inspection of the figures shows that all the maps hold similar results; most of the cattle movements happened within the same nodes. Nevertheless, there are a few pairs of nodes that are connected by *one step percolations*.

Only the January 2004 map (Figure 6.23) features a two step link. There were

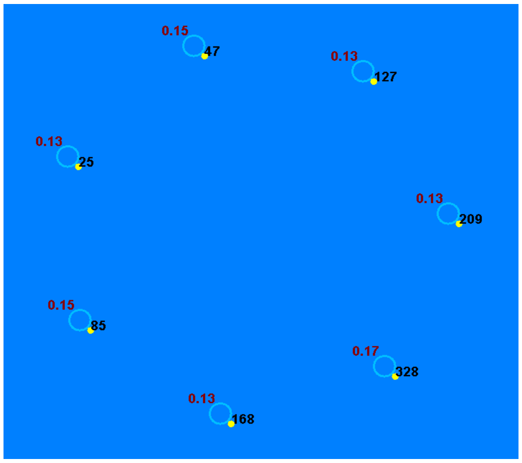


Figure 6.17: March 2003 Type 3 Combination Patterns Probability Map

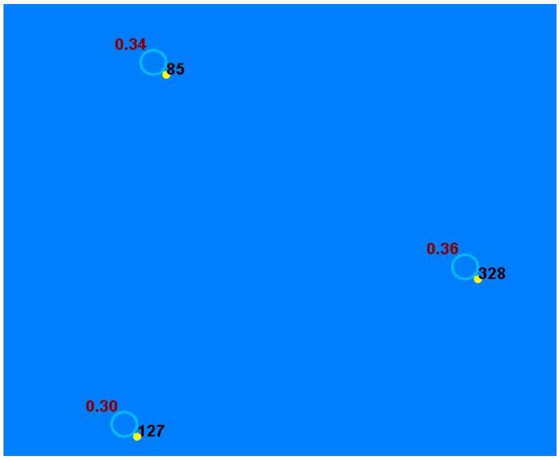


Figure 6.18: April 2003 Type 3 Combination Patterns Probability Map

movements that happen from within location area 127 to location area 148 with a probability of 0.0004 (0.02×0.02). Unlike the other islands on the map, if there is an event that happens within location 47 this may be caused by an event either in the location area 46 (with a probability of 0.02) or the location area 48 (with a probability of 0.02) or both with a probability of 0.0004.

6.5.4 Evaluation of the "Drill-down" Process With Respect To Specific Areas

Figures 6.26 and 6.27 show the outcome of the application of the "drill-down" process to location areas {127, 128, 147 and 148} for the prediction maps for January and February 2005 for Type 1 combination patterns. Notice that, originally there were four location areas (nodes) in the island that now are divided into 16 sub-areas. The number of cattle movements per area was therefore reduced and it is thus possible to identify which sub-areas the movements actually took place in. From the figures it is interesting to note (at least in this example) that the majority of cattle movements happened over distances of more than 25km but less than 50km. There are also movements within the 25km square areas. As mentioned in Sub-section 6.4, each node is divided in four sub nodes, for example $\{127(SW), 127(SE), 127(NE), 127(NW)\}$ which is labeled as $\{1271, 1272, 1273, 1274\}$ in the associated probability map.

Also, in both Figures 6.26 and 6.27 there were a number of two or more step percolations. For example, in Figure 6.26 an event (such as an outbreak of some cattle disease) occurring at location area 1482 could be transmitted to location area 1483 with a probability of 0.01, and to location area 1274 with a probability of 0.0001. In a three step percolation, in Figure 6.27, an event in location area 1272 could be transmitted to location area 1274 with a probability of 0.02, then to location 1283 with a probability of 0.0002 and to location 1284 with a probability of 0.000004.

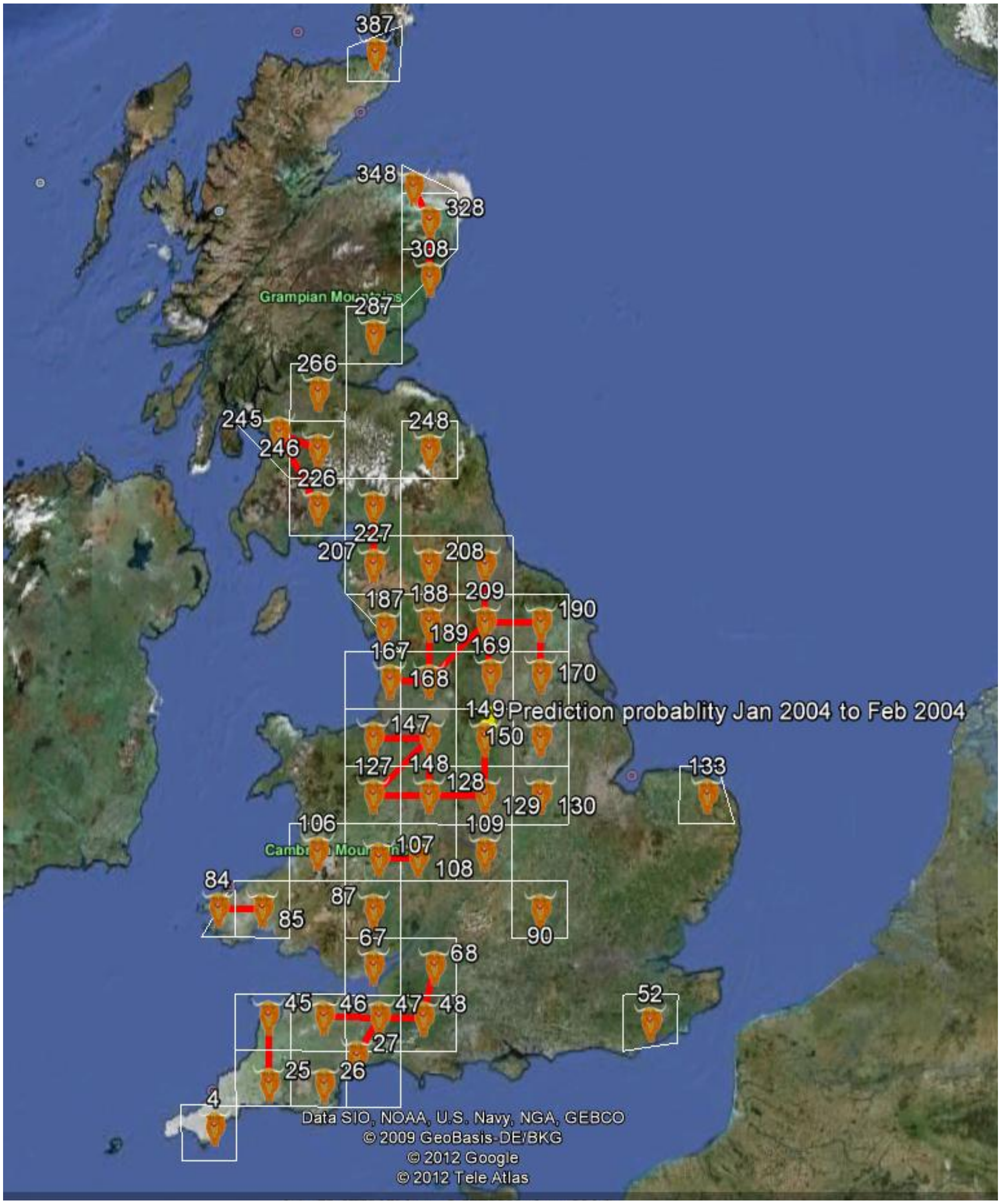


Figure 6.19: Location areas of January-February 2003 Type 1 Combination Patterns of Cattle Movement

6.6 Summary

This chapter has described the theory and operation of the Predictive Modeling modules that form the second part of the overall PTMF. The reported evaluation of the modules was conducted using the CTS database. The support values associated with particular kinds of frequent patterns was used to identify the probability of informa-

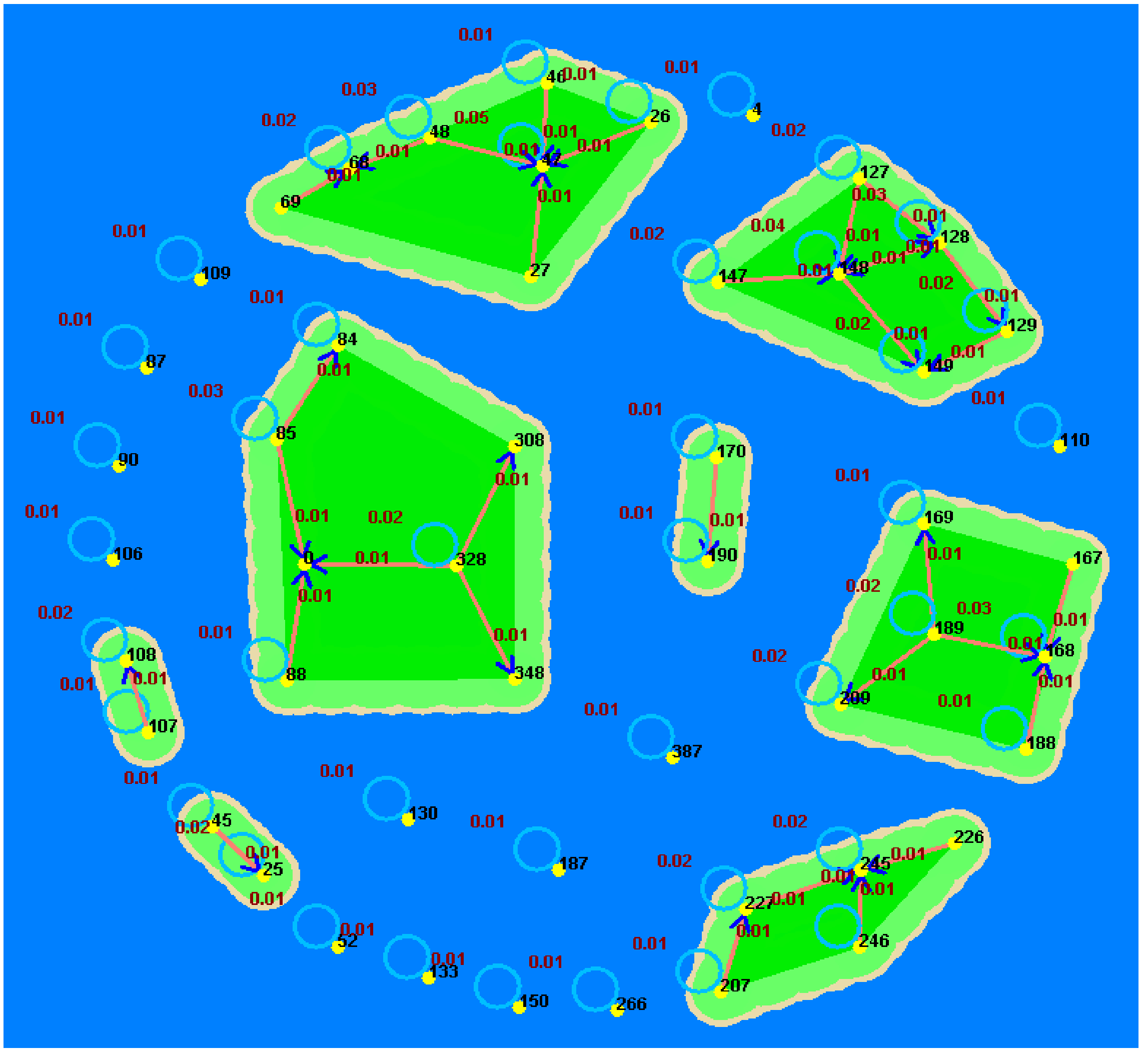


Figure 6.20: January 2004 Type 1 Combination Patterns Probability Map

tion from one node being transmitted to another node across a social network. The particular patterns of interest were those that comprise both node and link attributes (combination patterns). The CTS frequent patterns were generated using the Trend Identification module described previously. The proposed predictive modeling comprised two modules, the Percolation Matrix module and the vissualisation module. The first comprised two main processes: (i) filtering a specific type of combination pattern and (ii) converting the patterns' support values into probabilities and generating n probability percolating matrices. The Visualisation module provides two types of visualisation: (i) probability map visualisation and (ii) geographical map visualisation using Google Earth. The Prediction Modeling also provides an option to drill down into some selected parts of the probability map. The reported evaluation indicated how the overall process may be used to allow users to analyse networks and predict how an event or information may travel across a given networks. From the above, it should be noted that the Prediction Modeling produces a global probability prediction of an

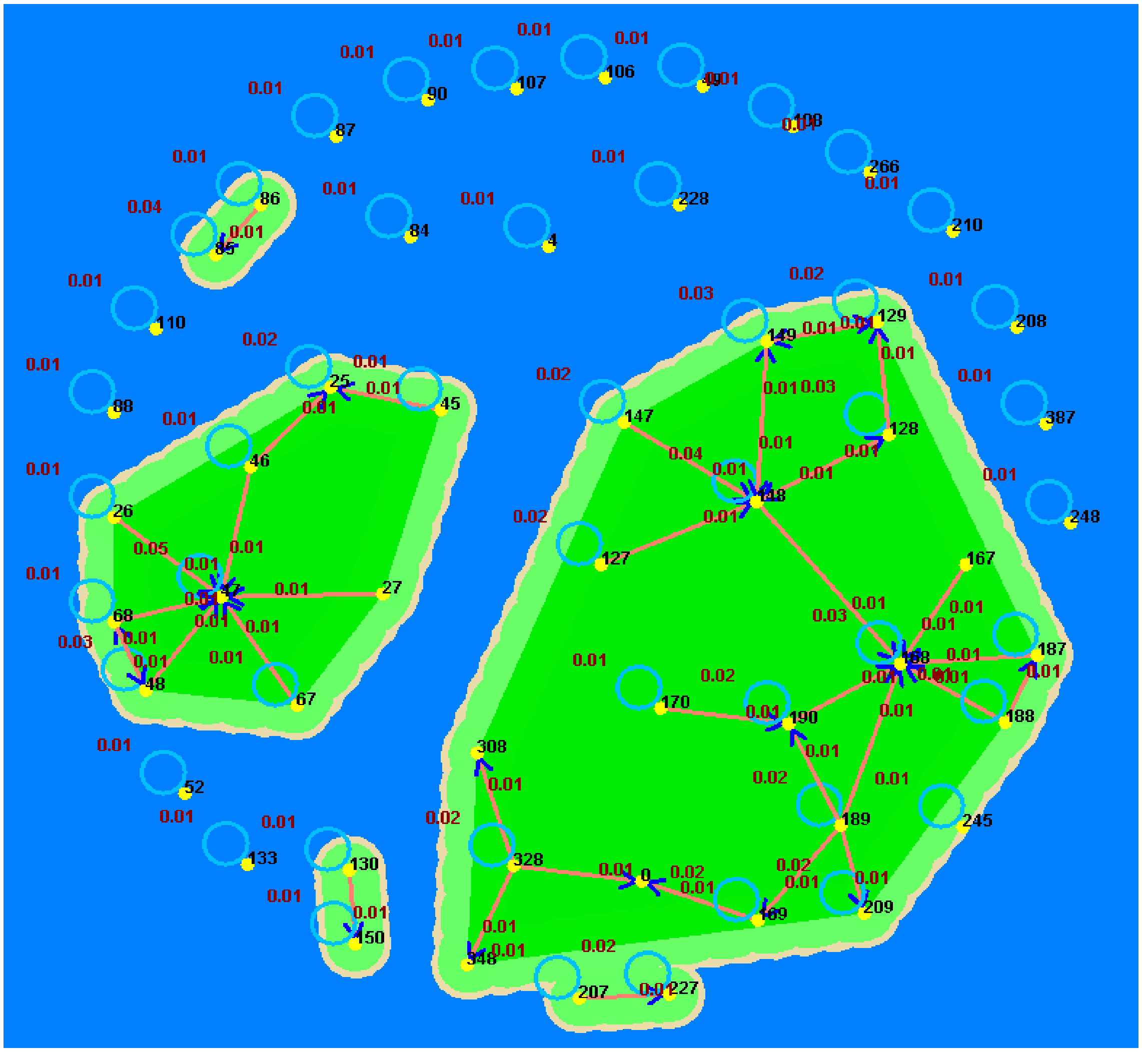


Figure 6.21: January 2005 Type 1 Combination Patterns Probability Map

event occurring at some node X be transmitted to node Y. For many applications this would be perfectly adequate, however for some application we might wish to ascertain the probability of an event occurring at a specific X being transmitted to Y. We will return to this issue in the next chapter, the concluding chapter of this thesis.

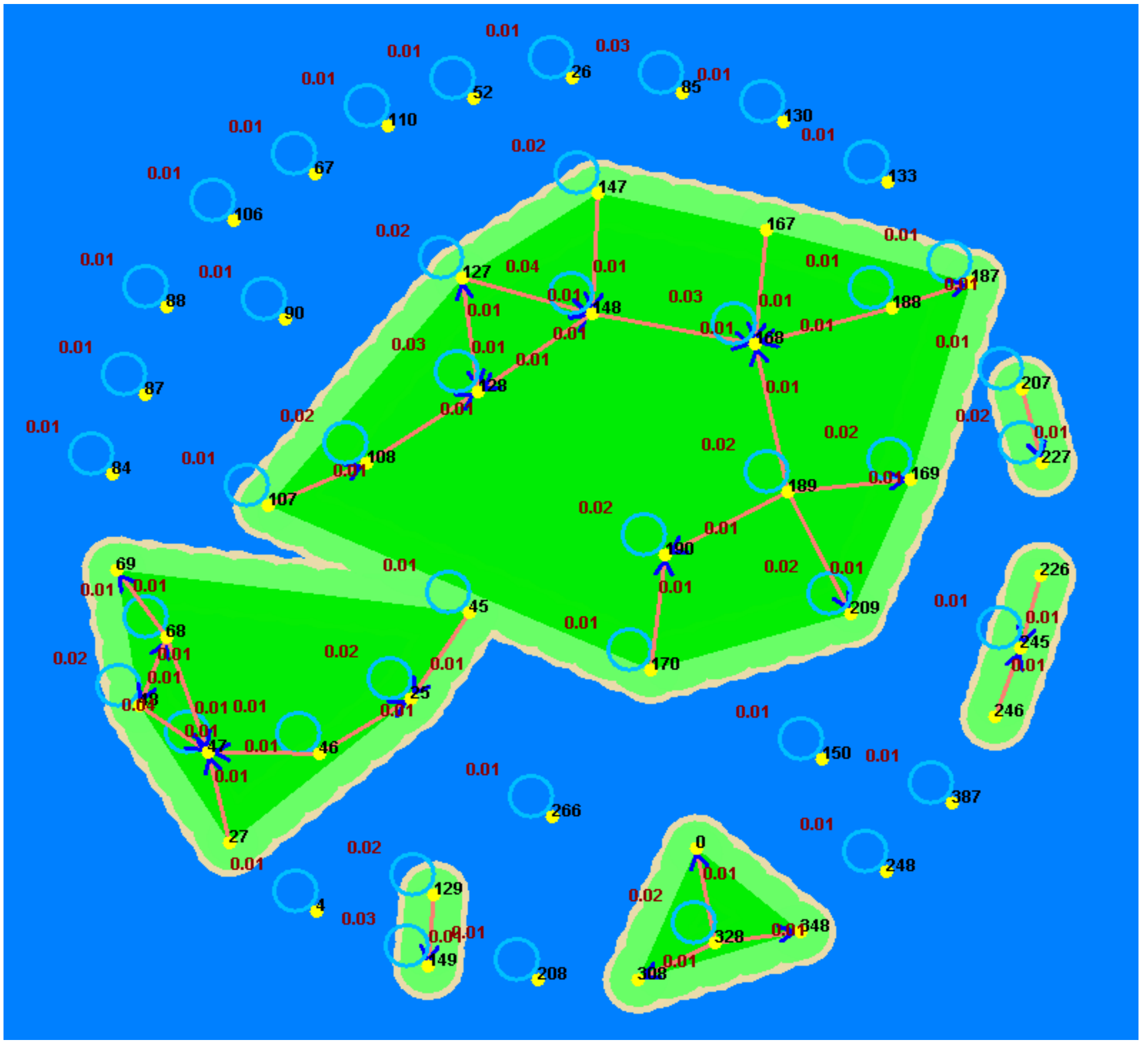


Figure 6.22: January 2006 Type 1 Combination Patterns Probability Map

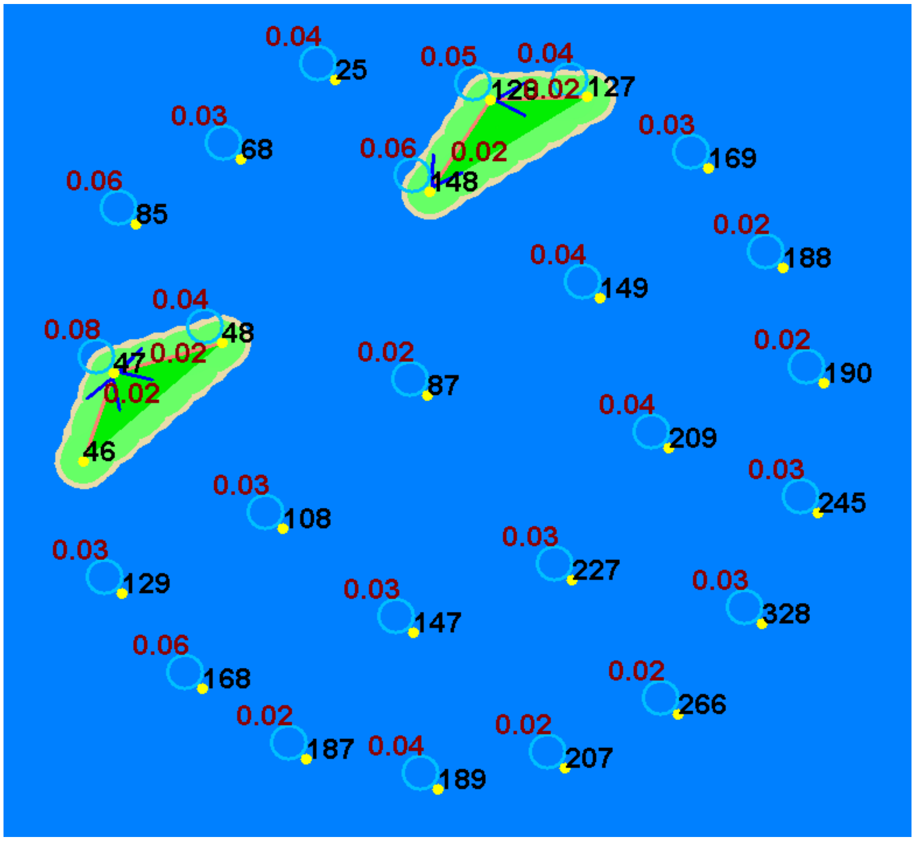


Figure 6.23: January 2004 Type 2 Combination Patterns Probability Map

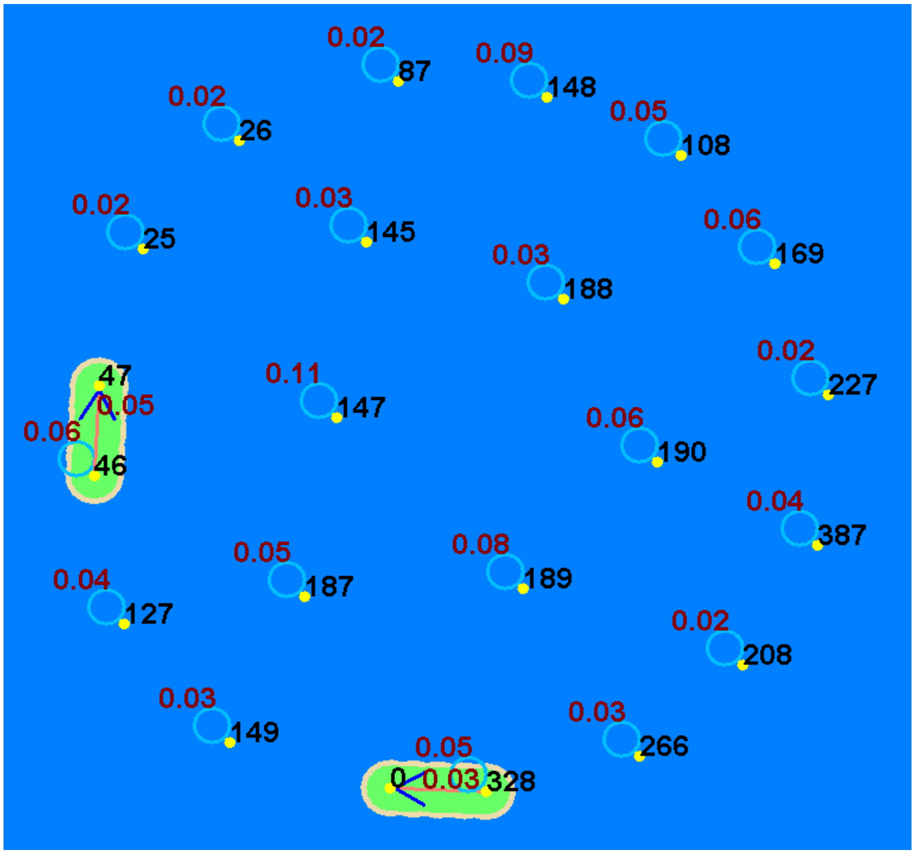


Figure 6.24: January 2005 Type 2 Combination Patterns Probability Map

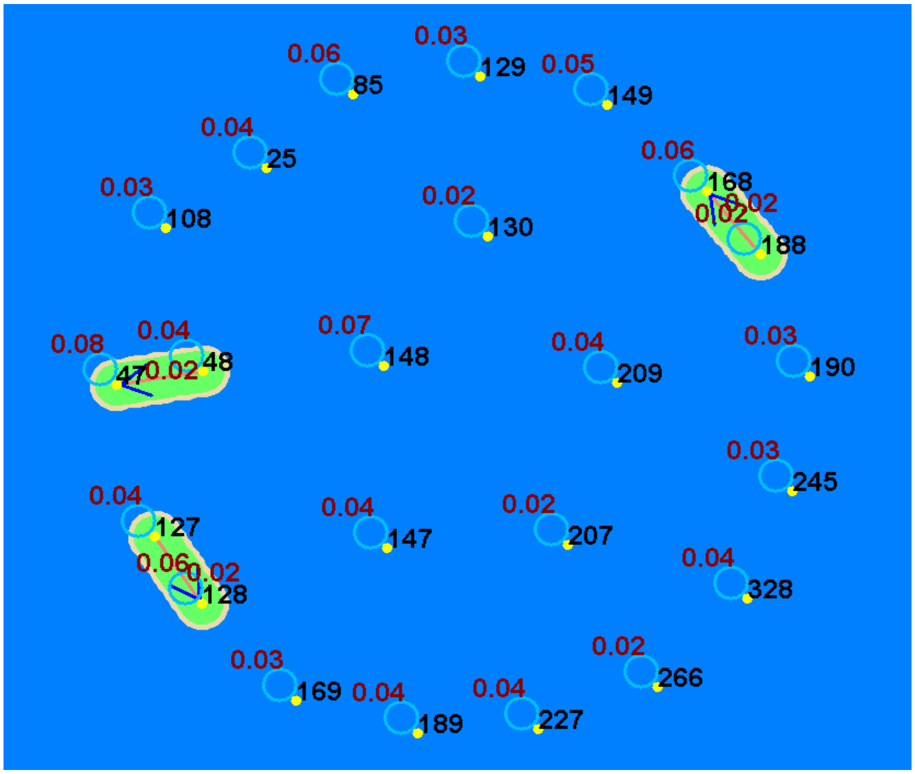


Figure 6.25: January 2006 Type 2 Combination Patterns Probability Map

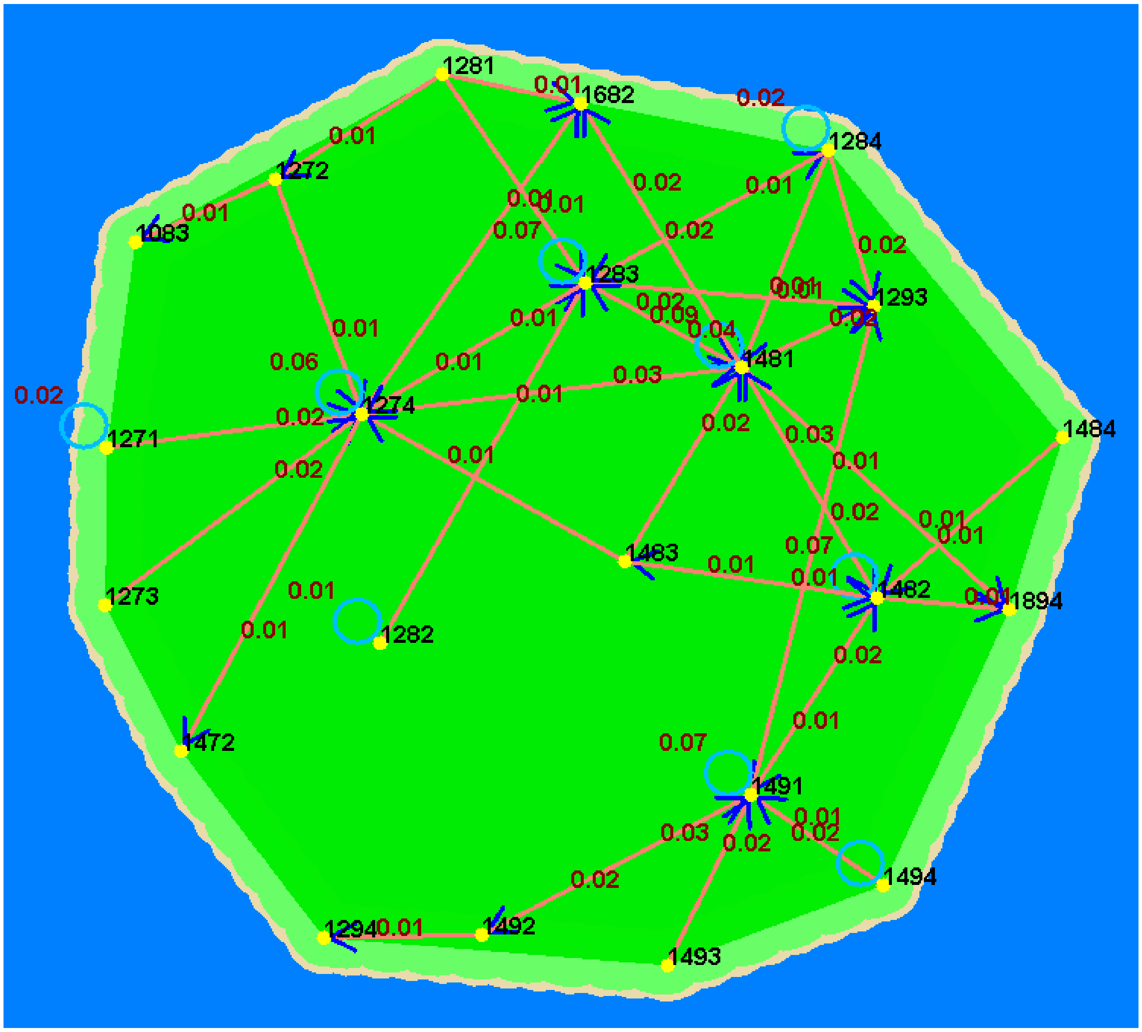


Figure 6.26: “Drill-down” version of January 2005 Type 1 Combination Patterns Probability Map

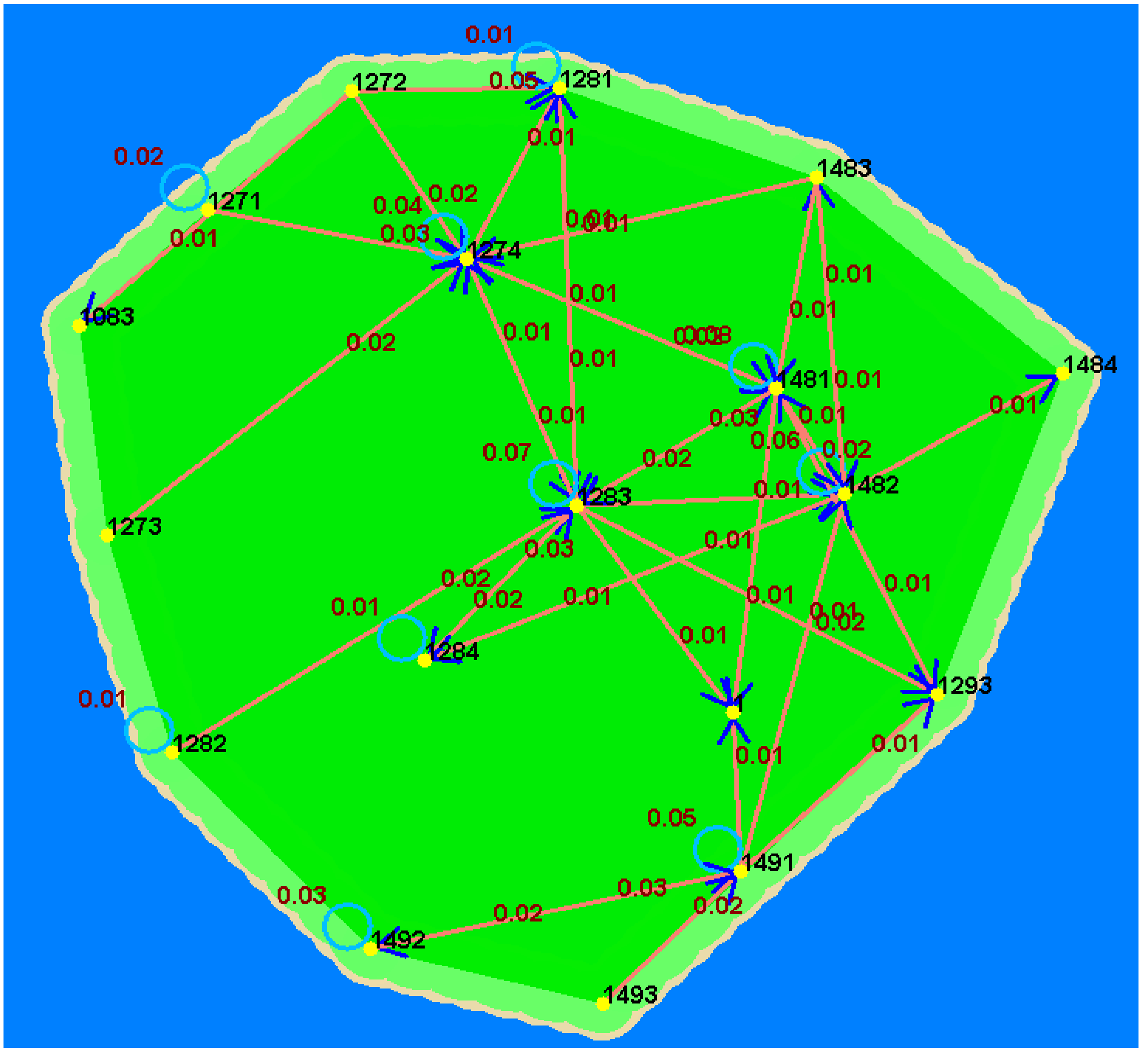


Figure 6.27: "Drill-down" version of February 2005 Type 1 Combination Patterns Probability Map

Chapter 7

Conclusion

The theme of this thesis has been trend mining. The view taken is that trend mining is a type of temporal data mining that provides insight into how information changes over time in the context of some activities. The idea is that knowledge and analysis of change will help organisations with respect to their strategic planning and operations management. The work described in this thesis was directed at mechanisms to not only identify change but also support the analysis and utilisation of change. To this end a number of data mining based technologies were investigated and proposed. These were combined into a single framework, called the Predictive Trend Mining Framework (PTMF) designed to support "end-to-end" trend mining and analysis. More specifically the thesis proposed a temporal frequent pattern mining algorithm to identify change expressed as trends, trend clustering and visualisation techniques to support trend understanding and analysis and an event prediction mechanism to support more advanced analysis. A summary of the proposed Predictive Trend Mining approaches, the main findings with respect to the identified research issues and question, the research contributions and possible future directions, are therefore presented in this chapter. Section 7.1 gives the summary of the proposed Predictive Trend Mining Framework and the main findings. The contribution of the research work, in relation to the research question and associated research issues identified in Chapter 1 are then presented in Section 7.2 and the research contributed reemphasised in Section 7.3. Finally some directions for future research are suggested in Section 7.4.

7.1 Summary

The objective of the proposed Predictive Trend Mining Framework (PTMF) is to benefit end users and stakeholders seeking to observe, analyse and possibly to take actions according to changing events occurring within their network environment of operation. The PTMF comprises two main parts: (i) Frequent Pattern Trend Analysis (FPTA) and (ii) Prediction Modeling (PM). FPTA is the process of identifying temporal frequent patterns and trends, and provides facilities to analyse these frequent pattern trends.

The FPTA element of the PTMF comprises four modules that are designed to be applied in order: (i) Trend Identification, (ii) Trend Grouping, (iii) Pattern Migration Clustering and (iv) Pattern Migration Visualisation. The Trend Identification module uses the TM-TFP algorithm to identify frequent patterns from a set of data episodes. One of the fundamental ideas promoted by the work described is the idea that a pattern trend can be defined in terms of a sequence of support values. This idea can then be extended to cover the concept of a related sequence of pattern trends describing a set of episodes. The analysis of the frequent pattern trends starts with the Trend Grouping module that clusters similar types of trends, using a SOM, to allow users to focus on trends of interest and "communities" of trend clusters. Further analysis directed at changes in the frequent pattern trends is facilitated by the Pattern Migration Clustering module. The interpretation of the pattern migration result is further facilitated by the Visualisation module. The second part of PTMF, the prediction modeling is designed to demonstrate how information or events may percolate across a (social) network. The Prediction Modeling consists of two modules: (i) the Percolation Matrix module and (ii) the Prediction Visualisation module. The first operates using the support associated with patterns to produce a Prediction matrix indicating the likelihood of how information may flow across a network form node to node. The second provides a visualisation of this Percolation Matrix.

The proposed framework has been evaluated using a number of time stamped social network datasets. The findings of the experimental analyses have shown that the PTMF serves to provide solutions to the main research issues and questions which were discussed in the Chapter 1. The evaluations, using different social network datasets, has also served to demonstrate the flexibility, reusability, genericity and accuracy of the PMTF.

7.2 Main Findings

As stated in Chapter 1, the key aim of the work described in this thesis is to establish and investigate effective mechanisms to: (i) discover temporal frequent patterns and trends in network data, and (ii) facilitate the analysis of these trends and patterns to predict behaviour across networks. In this section the findings of the reported experimental analyses are discussed in the context of the research issues central to this thesis:

1. **Frequent Patterns and Trends**: The identification of frequent patterns and trends using the TM-TFP algorithm allows users to discover hidden information in social network data. The discretisation and normalisation mechanism provides the conversion process for the raw data into the pre-processed (binary valued) input datasets so as to provide a flexible and reusable format for the proposed

Trend Identification module to overcome the different characteristics of the potential input data. To support large temporal data, the concept of individual episodes (time series) permits a collection of temporal patterns and trends to be generated for analysis purposes. The granularity of time stamps and selection of data feature (attributes) are subjective to the interest of the users, however flexibility is provided to allow user to define the nature of the data episodes and features. This also allows the mining and analysis process to focus and highlight the time series results according to users' interest.

2. **Change Detection**: With a sufficiently large number of discovered temporal patterns and trends, temporal changes in the trends can be identified to support further trend analysis. In the thesis this is conceived of in terms of "pattern migrations". The proposed mechanism whereby these migrations can be identified, by comparing SOM maps, clearly provides for the desired detection of temporal changes in the network data.

3. **Interesting Trends**: Interesting tends are defined as those that migrate in some way. The further they migrate the more interesting they are deemed to be. The proposed mechanism to support the identification of interesting migrations represents a much more sophisticated technique for defining interestingness than a simple support thresholding technique. In addition, constraints can be applied to filter data records depending on the nature of the user's interest. When identifying pattern migrations, a minimum distance threshold is used to determine the interestingness or significance of the migration. This threshold can be adjusted so that the user can identify the most significant migrations.

4. **Interpretation of Patterns and Trends**: The clustering facility, using a SOM, provides a mechanism for supporting the analysis of trends by grouping similar trends together. The trend grouping (clustering) allows users to identify types of trends that exist in a set of network data episodes. In addition the migration visualisation module illustrates how pattern migrations may occur in a way that is readily accessible to end users.

5. **Prediction**: The proposed Percolation Matrix supports the idea of predicting how events might "percolate" across a network. The Percolation Matrix is constructed using frequency counts. The conversion of the identified trend information into probability values indicates the likelihood of events percolating (travelling) between location patterns with respect to time. The mechanism also provides information concerning both direct and indirect "percolation paths".

6. **Visualization**: The proposed Pattern Migration Visualisation module was used to display the movement across pairs of pattern migration maps describing two

subsequent data episodes. The visualisation benefits the users in that it allows for better interpretation of the result of the trend cluster analysis. In the case of the Prediction Modeling Visualisation module, the information in the percolation matrices is displayed. Users are thus able to view the possible percolation paths. The maps also indicate the probability of particular movements. In the case of (say) infection spread, users can identify both the possible source and the final destination of the infection so that preventative action can be taken and future monitoring planed. The application of the Geographical map tool (supported with Google Earth) allows the identified percolations to be illustrated against an actual geographical "backdrop".

Thus, given the above findings, the proposed mechanisms, incorporated into the PTMF, can be said to addressed the principal research question which was: *"What are the most appropriate mechanism for identifying analyzing and displaying trends in network data and how might those trends be usefully be employed for prediction purposes?"* Referring to Section 1.4 in Chapter 1, the PTMF has also been evaluated to ensure that the framework is a quality and effective technique for trend mining and analysis, and prediction modeling. With respect to the research issues identified in Chapter 1:

- **Genericity**: The PTMF was designed to accommodate pre-processed binary valued data as this was considered to be a very general format that would support the processing of a variety of different kinds of social network data.
- **Computational time and memory**: For each experiments using the PTMF reasonable run times were recorded. The memory resource used to process and store the frequent pattern trends was also found not to be excessive.
- **Flexibility and Reusability**: Different types of social network data have been used to evaluate the PTMF indicating that the PTMF is both a reusable and flexible process.
- **Scalability**: The PTMF is able to process large dataset (such as the CTS dataset) and small datasets (such as MAF Logistic Cargo dataset).
- **Accuracy**: Analysis of the identified patterns and trends produced during the evaluations, with the support of domain experts, indicated that the correct patterns were identified and displayed.

7.3 Research Contributions

With respect to Section 1.5, the main contributions of the research work considered in this thesis can be summarized as follows:

1. A mechanism for efficiently generating temporal spatial frequent patterns and trends to identify patterns and trends within social networks.

2. A mechanism for clustering large numbers of trends, using a SOM technique, so as to assist in the further analysis of the identified trends.

3. A trend cluster analysis mechanism to support the detection of temporal changes in trends and frequent pattern migrations.

4. A visualization of pattern migrations (traffic) from one trend cluster to another over a period of time, again to facilitate and support trend analysis.

5. A mechanism for prediction modeling that can be applied to network data using the discovered frequent pattern trends, which illustrates the probability with which information (events) might travel across a social network.

7.4 Research Future Direction

A sound foundation to support trend mining and analysis has been established and incorporated into the PTMF. Nevertheless, there are a number of areas which merit further investigation so as to enhance the functionality and increase the overall quality the framework. The work described in this thesis has raised a number of promising directions to enhance the operation of the PTMF as follows:

- **Frequent pattern trends that fall below the support threshold**: The TM-TFP algorithm prunes the patterns that occurred below a specified support threshold. Thus, in certain time stamps, when the pattern happens to be infrequent, the patterns' trends are assumed to have a "0" value as opposed to the actual frequent count. This is because the counts are not stored in the P-trees. It would be desirable for the TM-TFP algorithm to be able to retrieve or store the actual frequency counts for any pattern that was frequent in the previous time stamps. Of course this should be done in an efficient manner with a good use of memory space.

- **SOM grid configurations**: There is currently no scientific method to determine an optimum SOM grid configuration, The identification of mechanisms to identify optimum SOM grid configurations would provide for more effective trend analysis and prediction modelling. This would also be of interest to the wider research community.

- **Trend Grouping module computational time**: The current computational time required by the Trend Grouping algorithm is significant, especially as the number of trends increases. The current system struggles to process large numbers of trends (in excess of 100,000). Better mechanisms and storage structures for the storing and processing of trends are therefore desirable so that larger networks and/or greater numbers of trends can be considered.

- **Predicting link traffic between a pair of specific nodes**: The current global prediction can be further investigated to propose a prediction modeling mechanism that can analyse how information or events travel from a specific node X to other specific node Y.

Appendix A

Probability Maps for CTS Type 1 Combination Patterns between June and December 2003

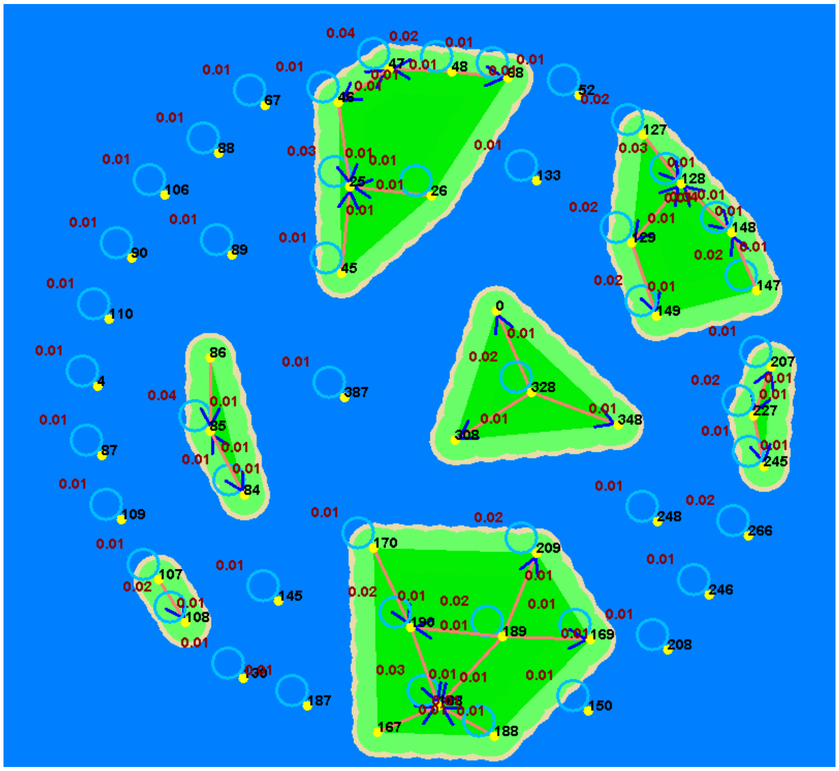


Figure A.1: June 2003 Type 1 Combination Patterns Probability Map

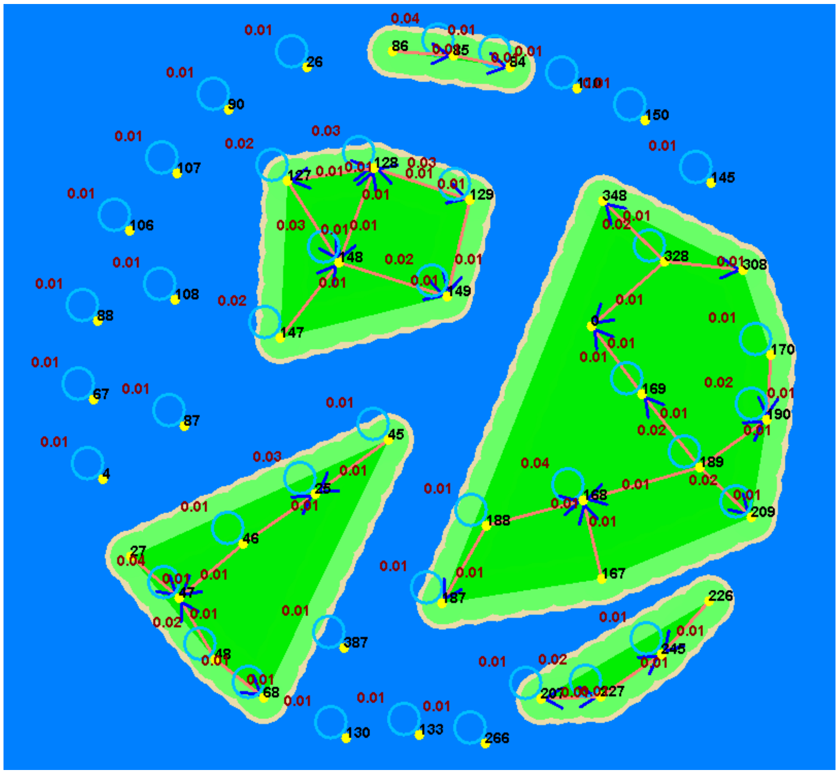


Figure A.2: July 2003 Type 1 Combination Patterns Probability Map

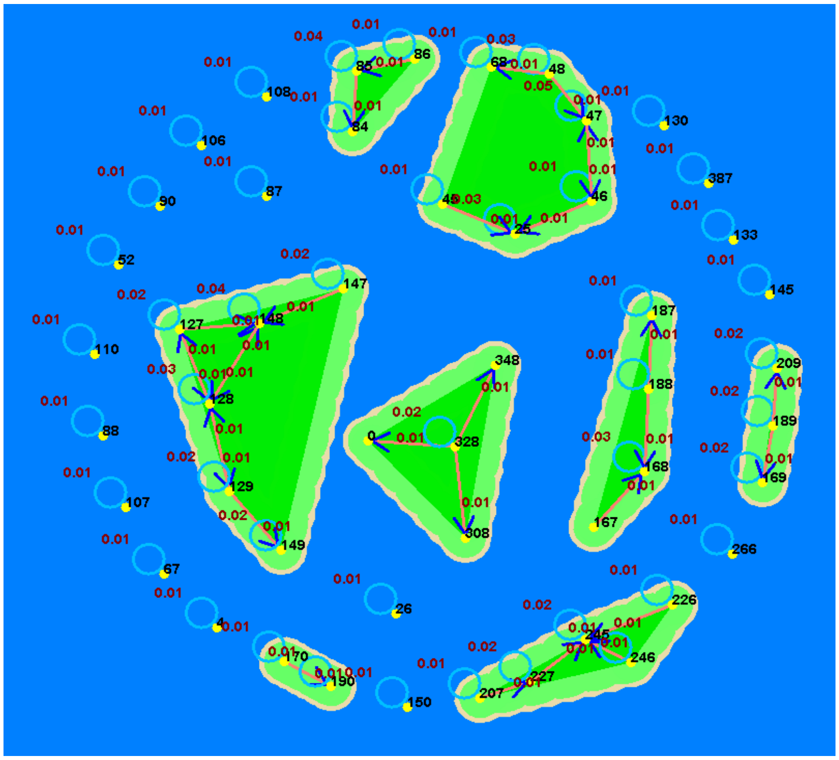


Figure A.3: August 2003 Type 1 Combination Patterns Probability Map

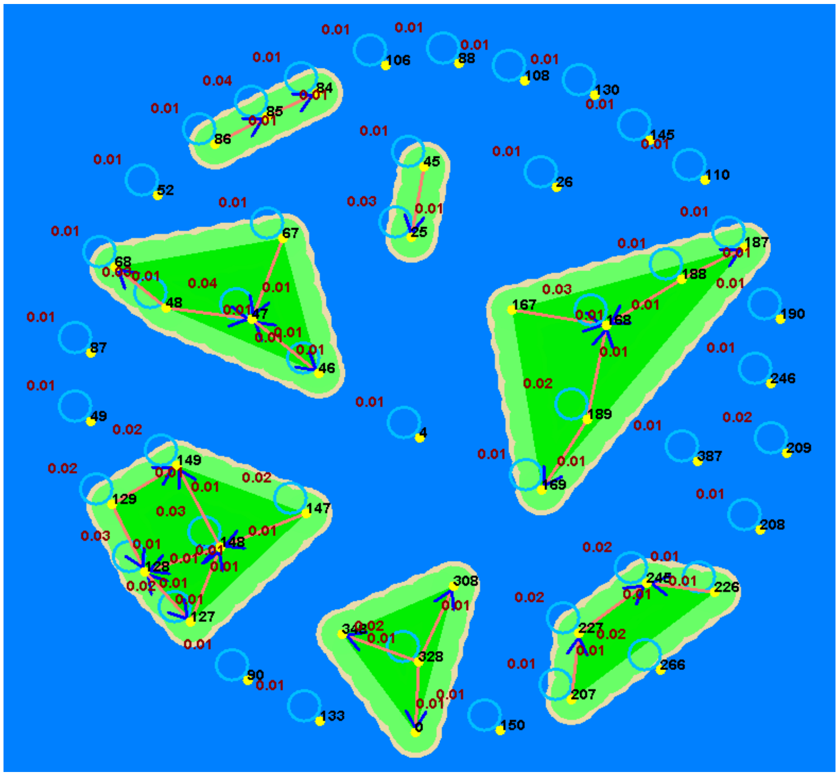


Figure A.4: September 2003 Type 1 Combination Patterns Probability Map

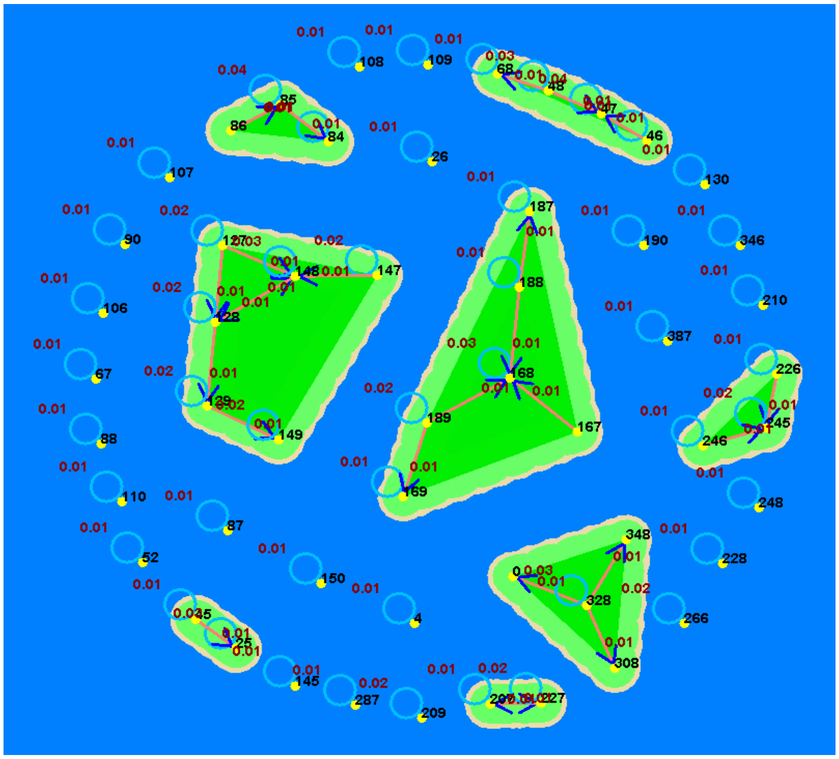


Figure A.5: October 2003 Type 1 Combination Patterns Probability Map

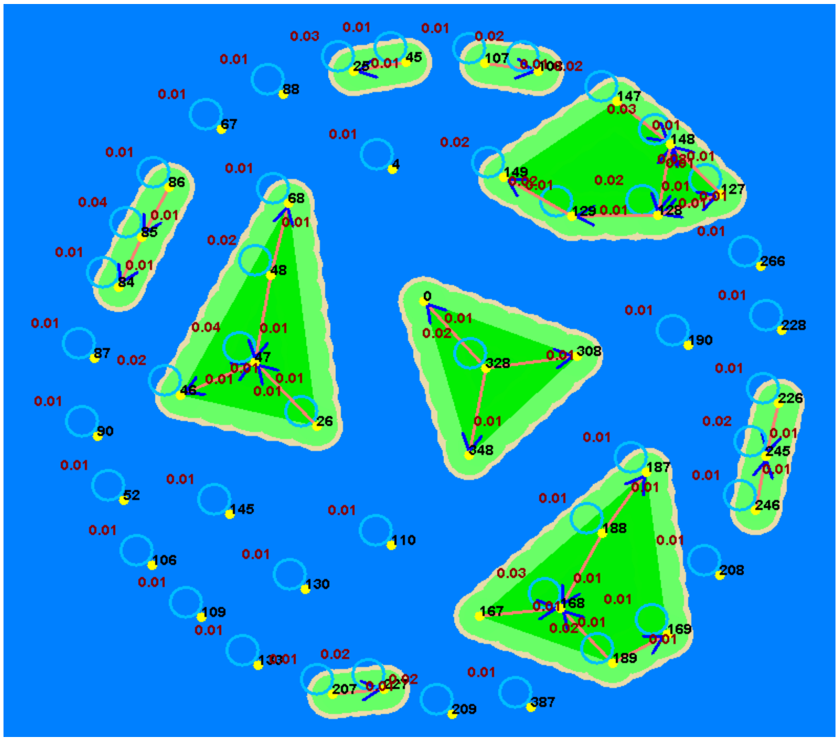


Figure A.6: November 2003 Type 1 Combination Patterns Probability Map

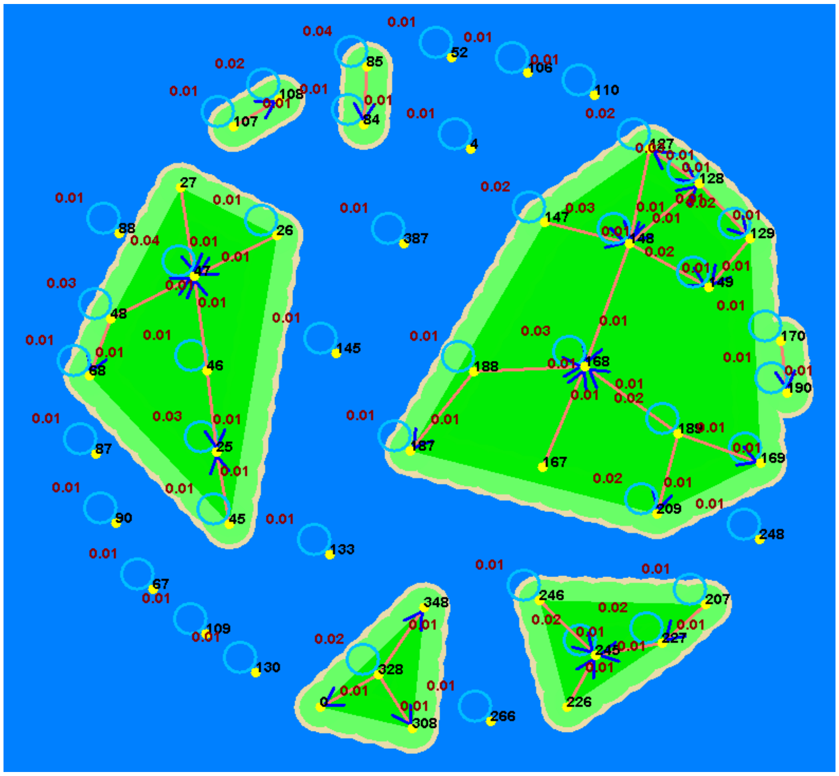


Figure A.7: December 2003 Type 1 Combination Patterns Probability Map

Appendix B

Probability Maps for CTS Type 1 Combination Patterns between February and December 2004

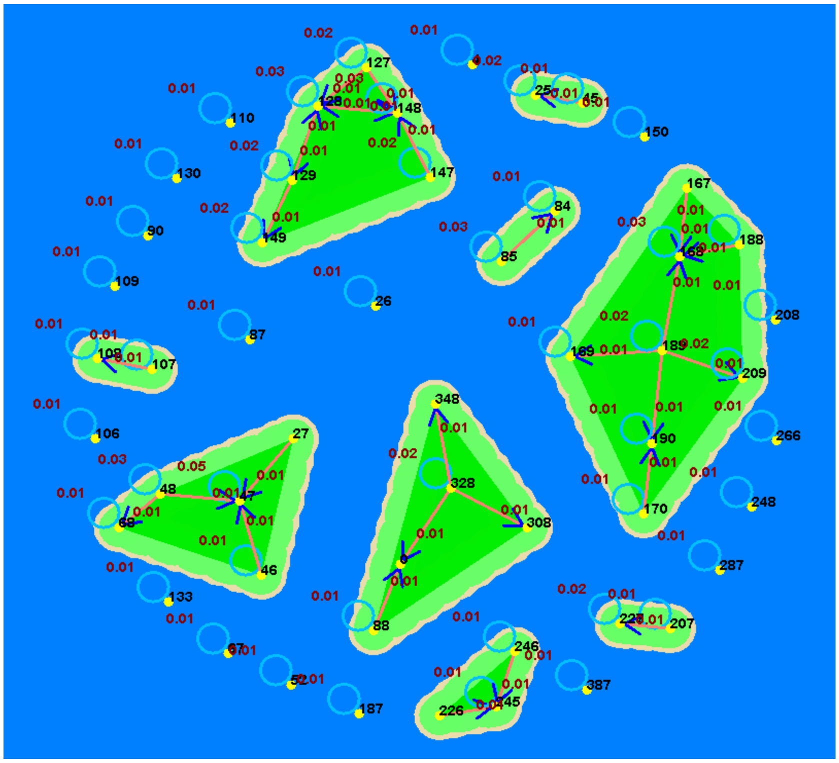


Figure B.1: February 2004 Type 1 Combination Patterns Probability Map

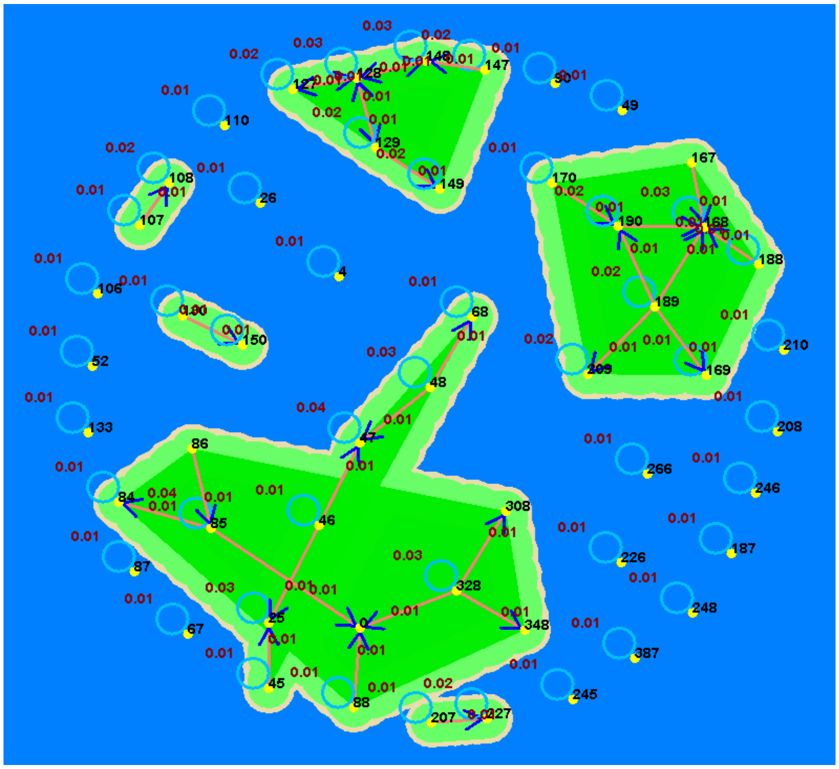


Figure B.2: March 2004 Type 1 Combination Patterns Probability Map

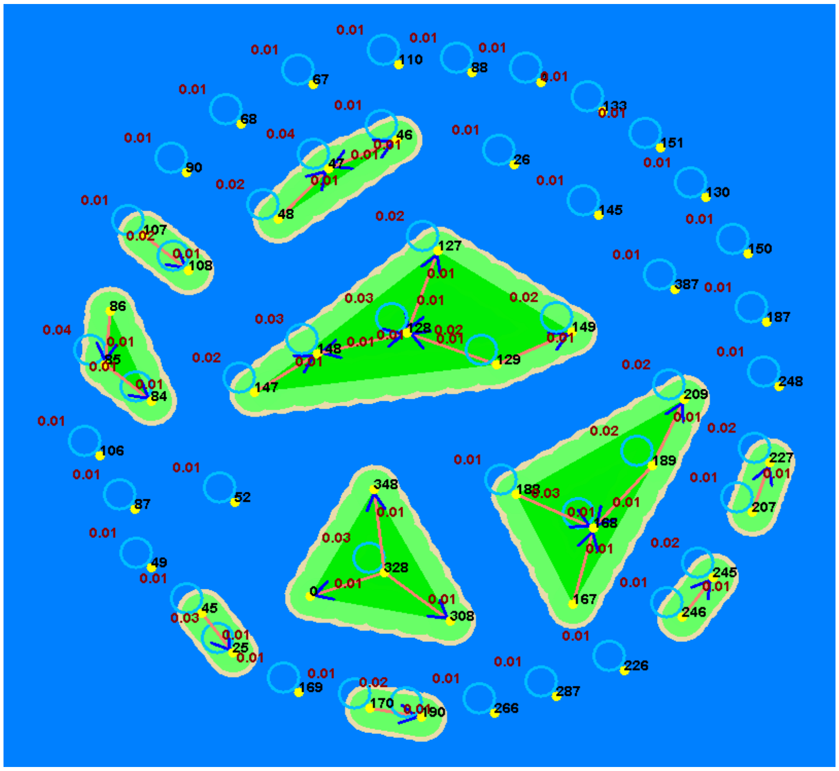


Figure B.3: April 2004 Type 1 Combination Patterns Probability Map

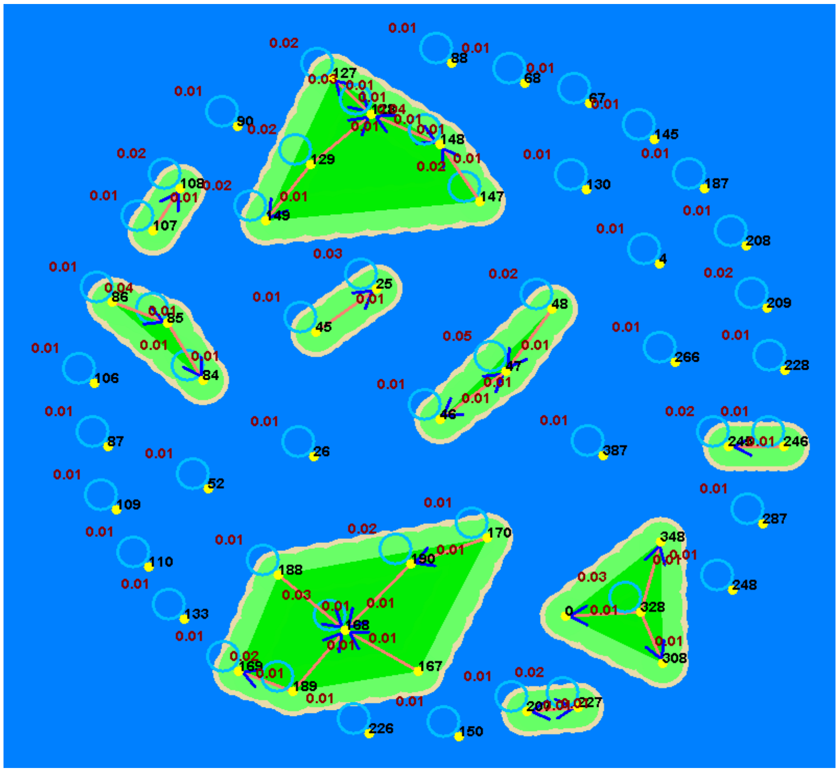


Figure B.4: May 2004 Type 1 Combination Patterns Probability Map

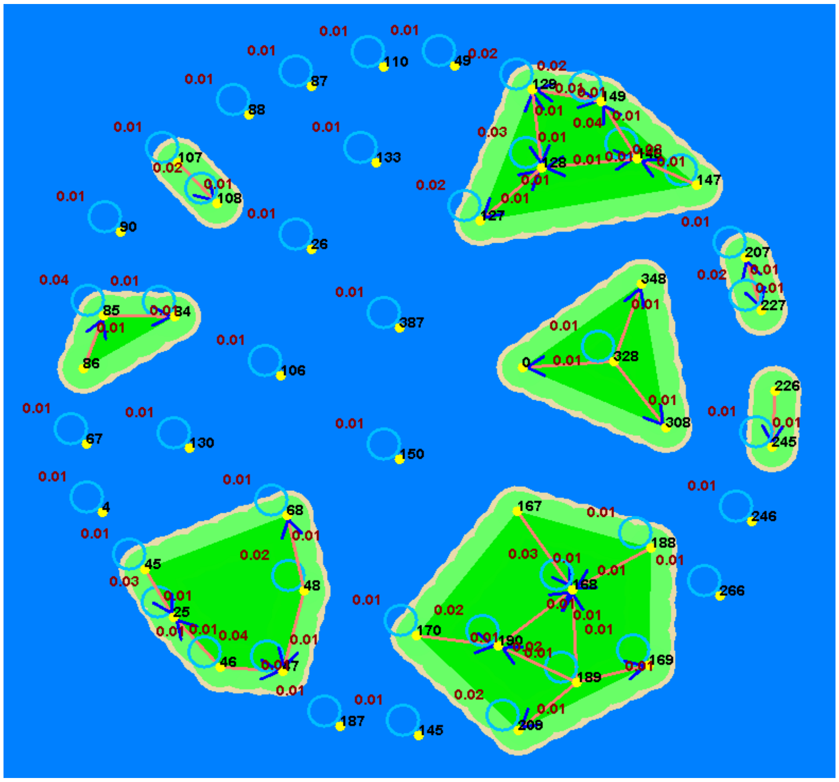


Figure B.5: June 2004 Type 1 Combination Patterns Probability Map

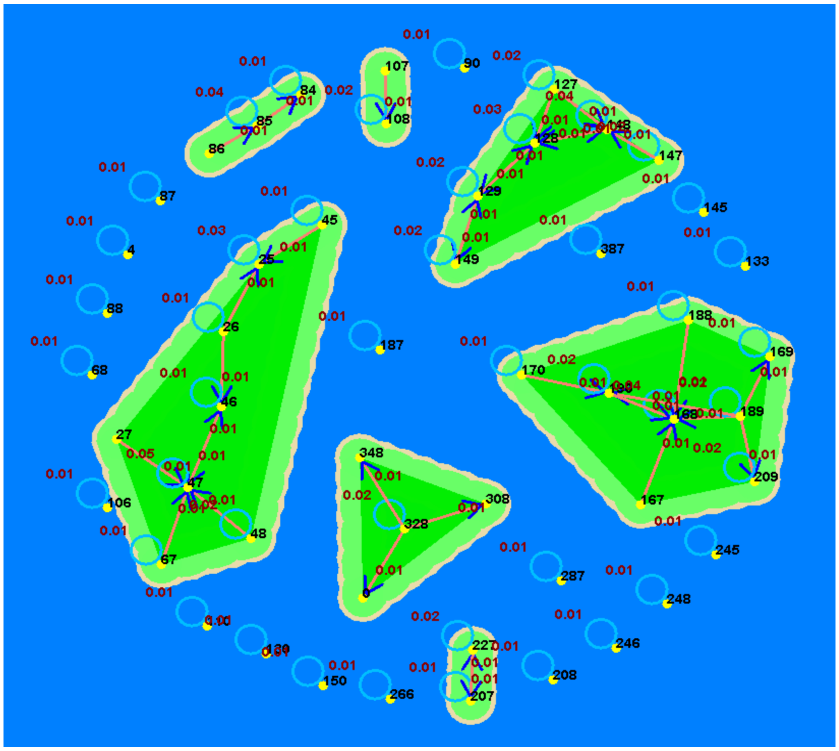


Figure B.6: July 2004 Type 1 Combination Patterns Probability Map

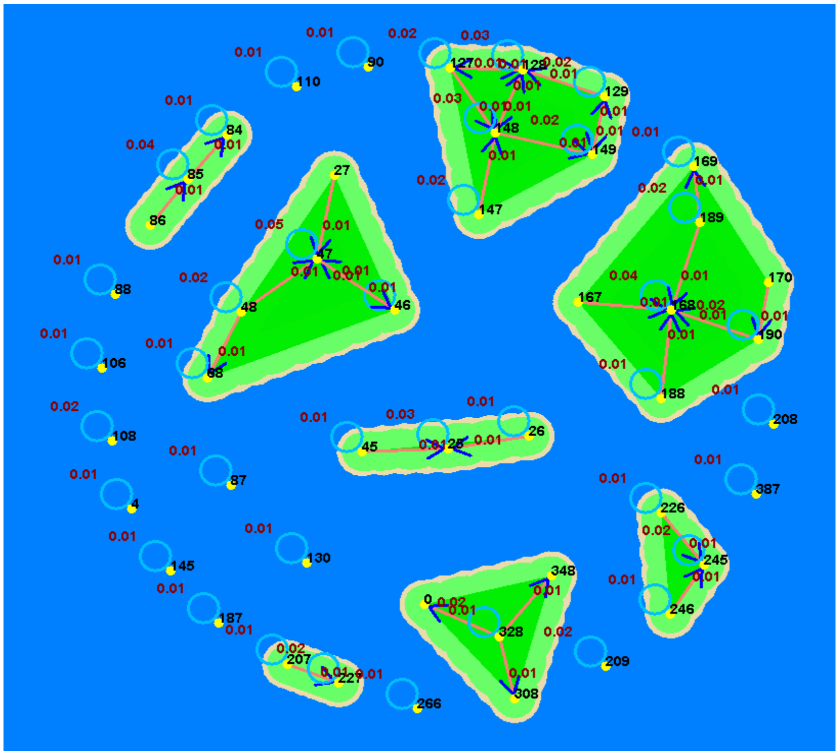


Figure B.7: August 2004 Type 1 Combination Patterns Probability Map

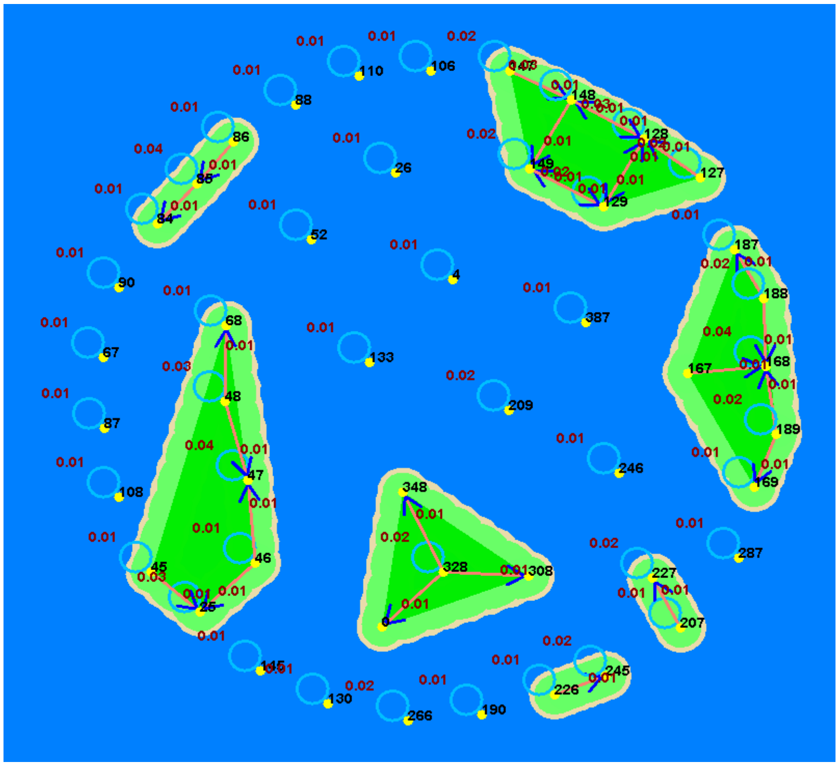


Figure B.8: September 2004 Type 1 Combination Patterns Probability Map

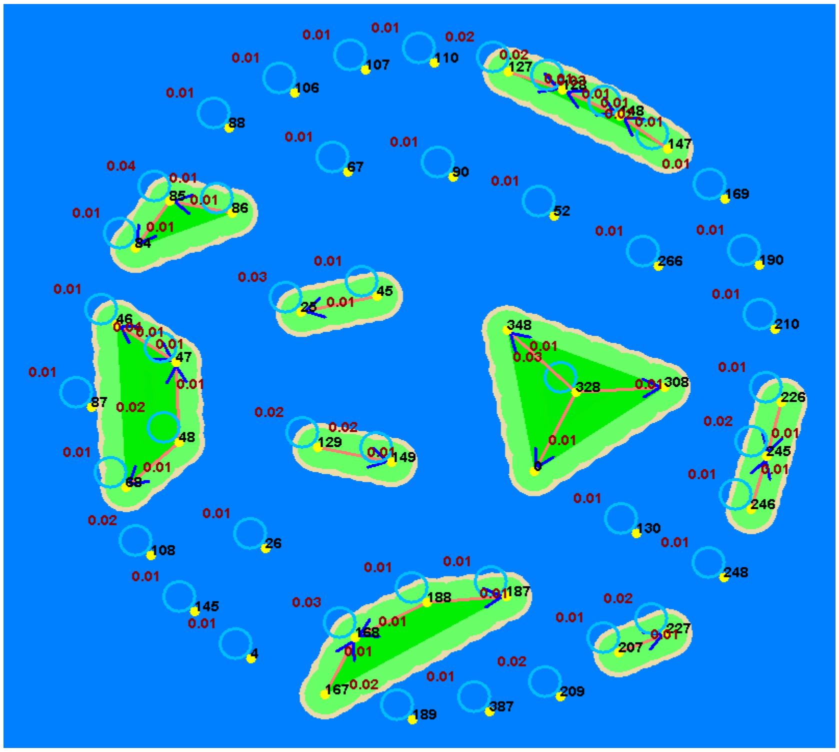


Figure B.9: October 2004 Type 1 Combination Patterns Probability Map

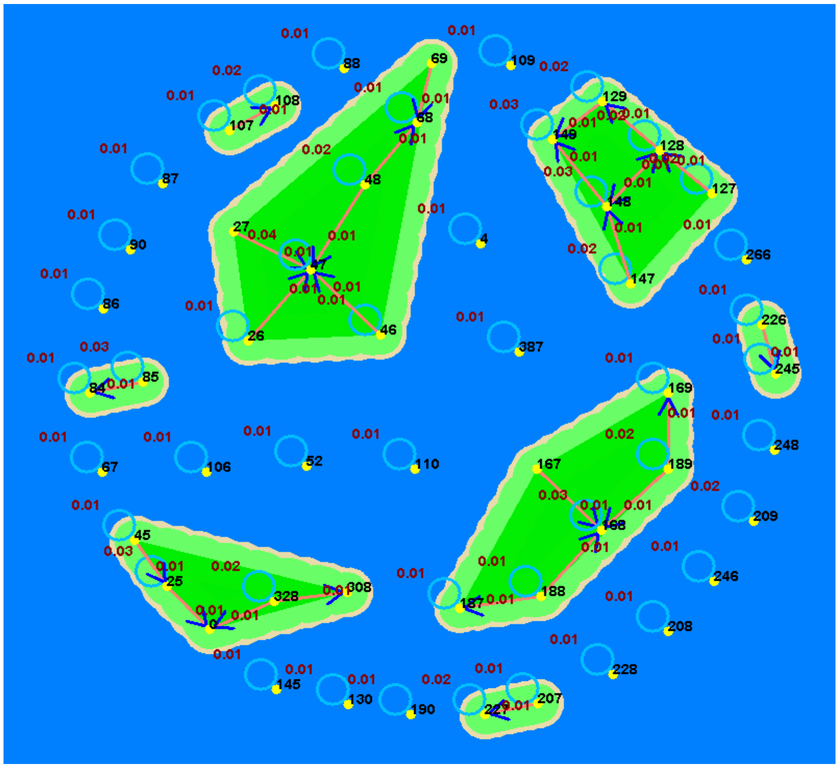


Figure B.10: November 2004 Type 1 Combination Patterns Probability Map

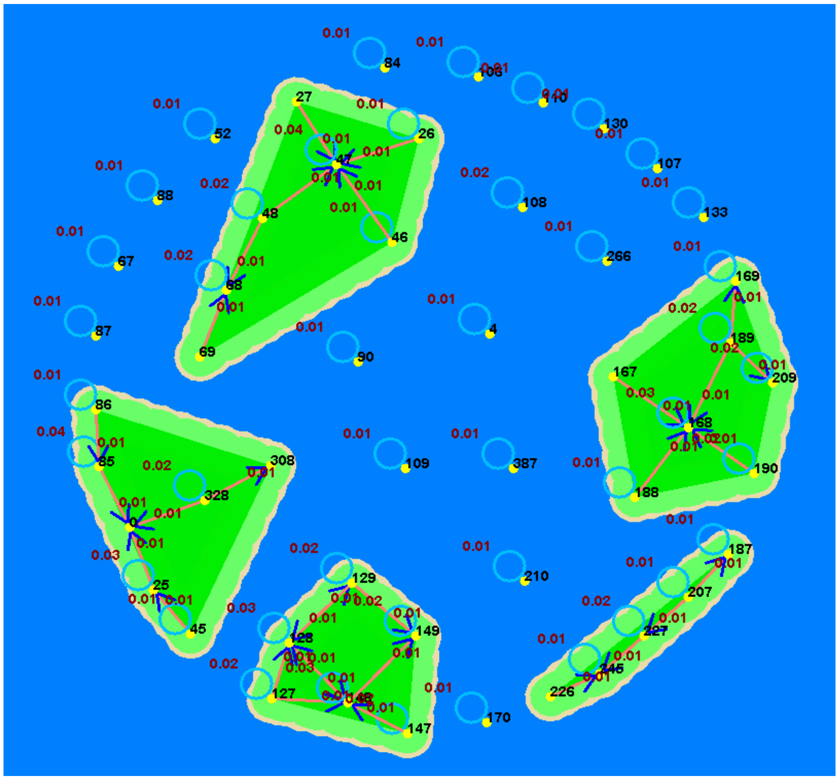


Figure B.11: December 2004 Type 1 Combination Patterns Probability Map

Appendix C

Probability Maps for CTS Type 1 Combination Patterns between February and December 2005

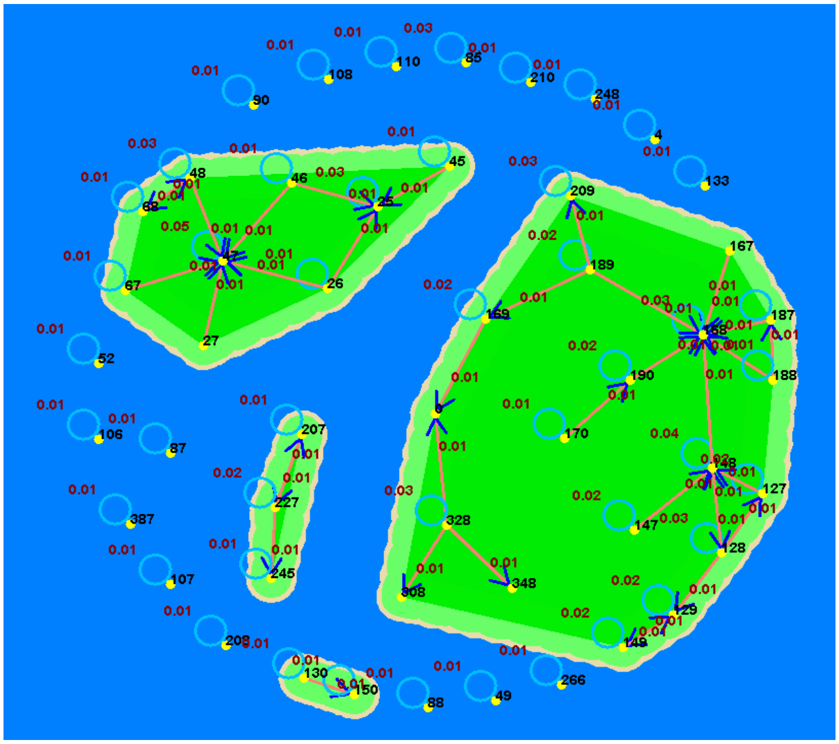


Figure C.1: February 2005 Type 1 Combination Patterns Probability Map

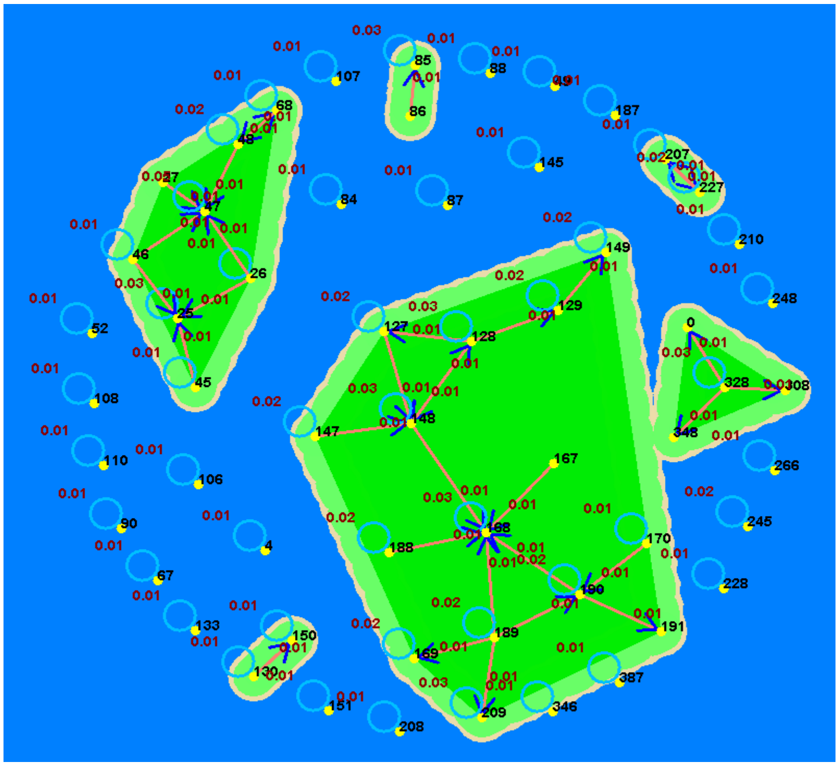


Figure C.2: March 2005 Type 1 Combination Patterns Probability Map

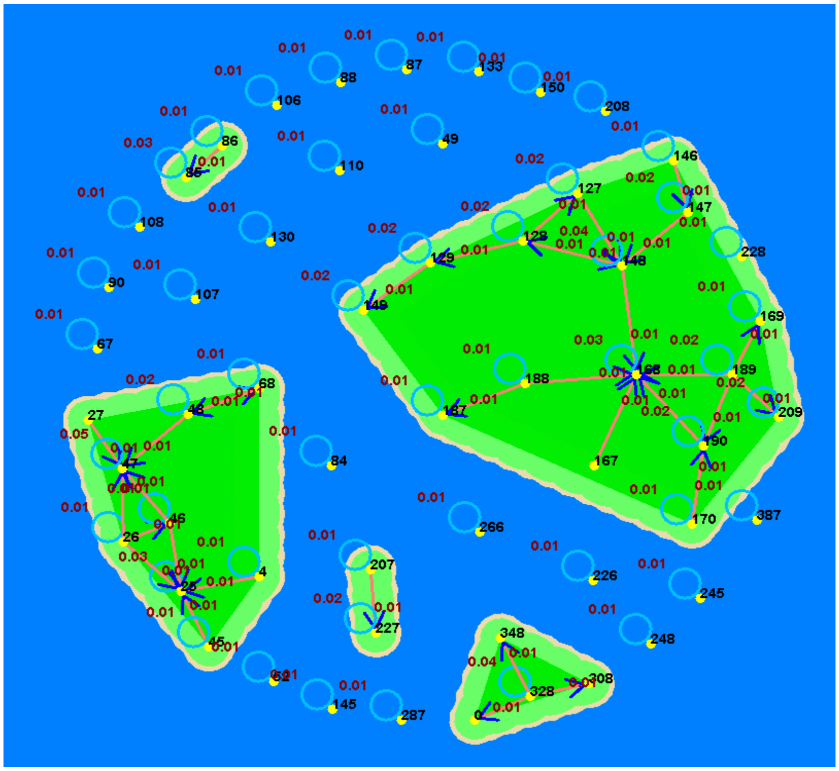


Figure C.3: April 2005 Type 1 Combination Patterns Probability Map

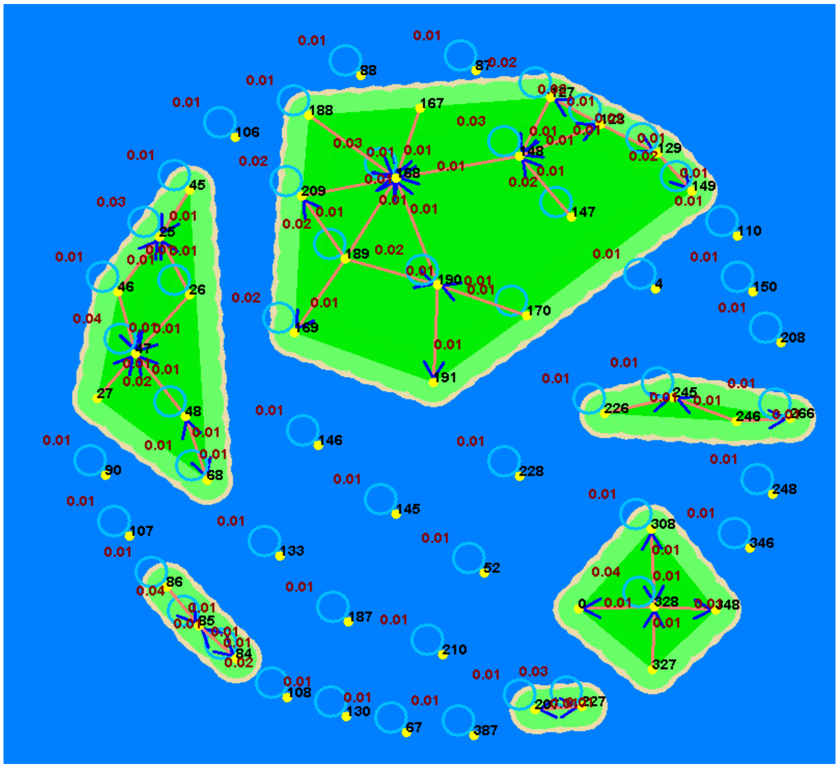


Figure C.4: May 2005 Type 1 Combination Patterns Probability Map

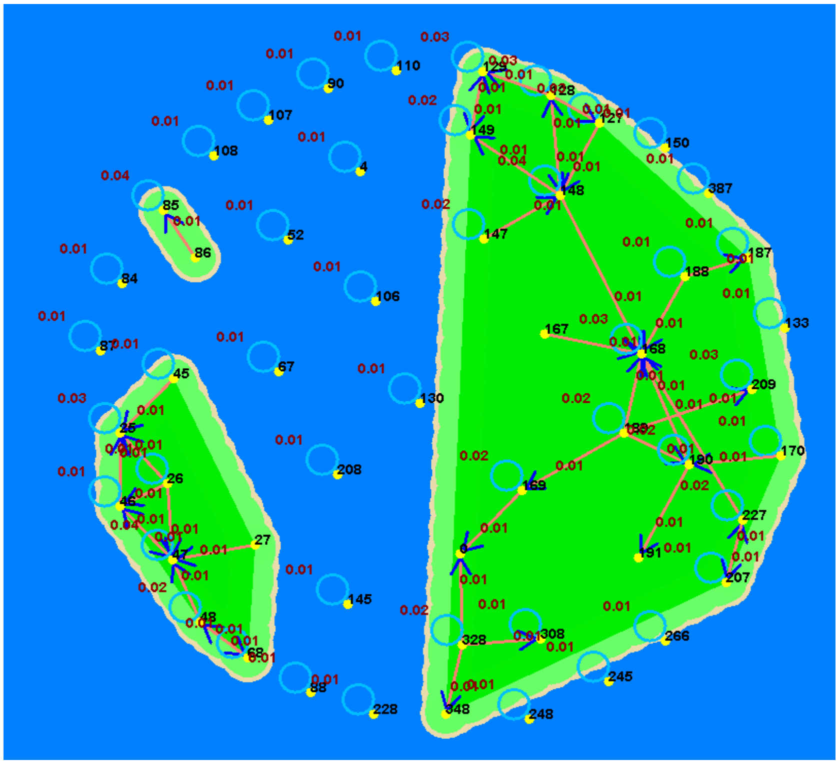


Figure C.5: June 2005 Type 1 Combination Patterns Probability Map

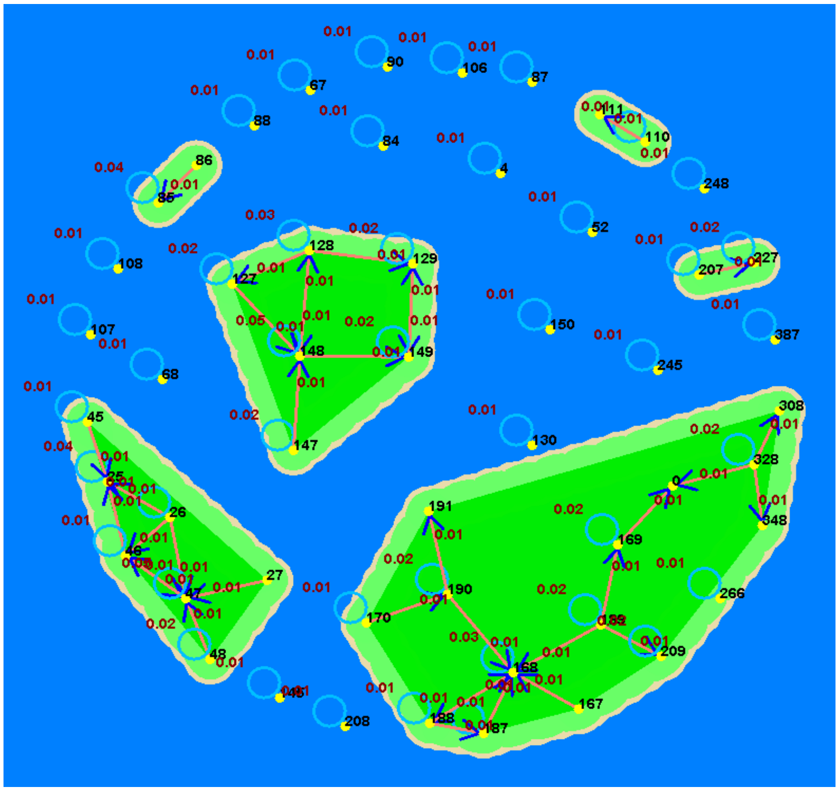


Figure C.6: July 2005 Type 1 Combination Patterns Probability Map

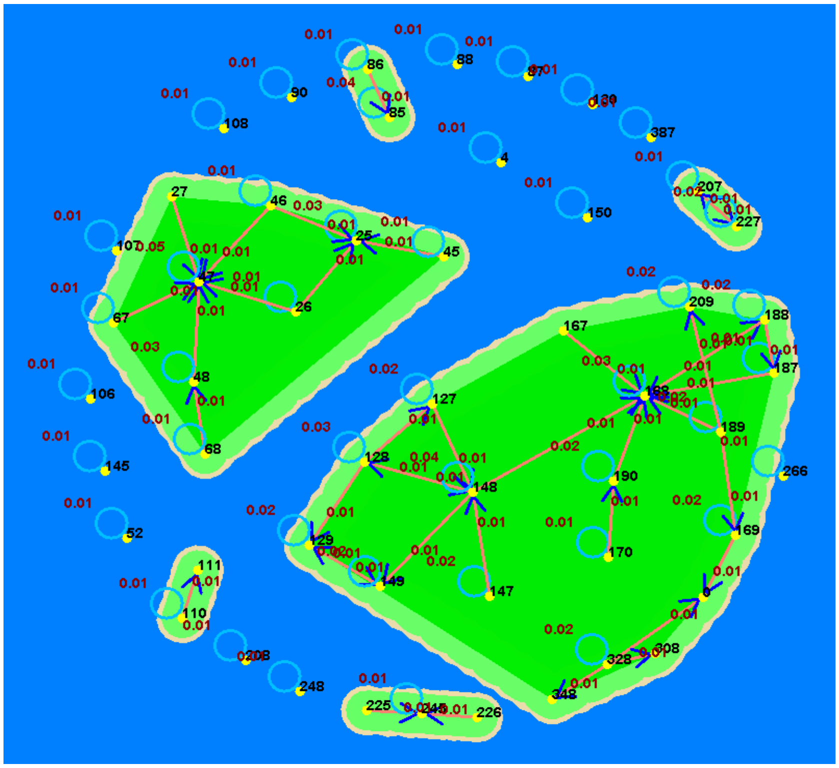


Figure C.7: August 2005 Type 1 Combination Patterns Probability Map

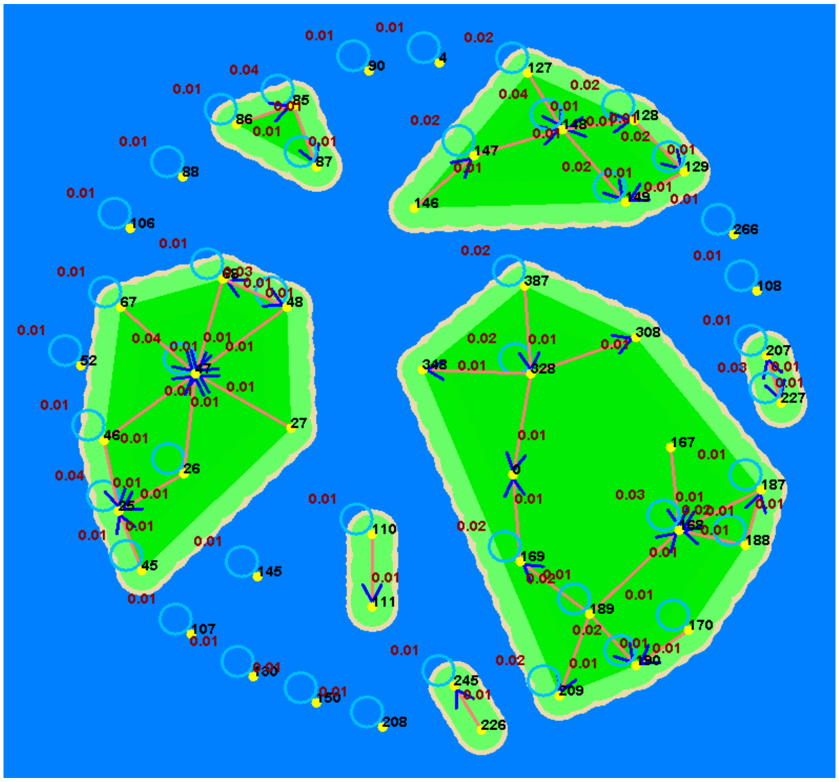


Figure C.8: September 2005 Type 1 Combination Patterns Probability Map

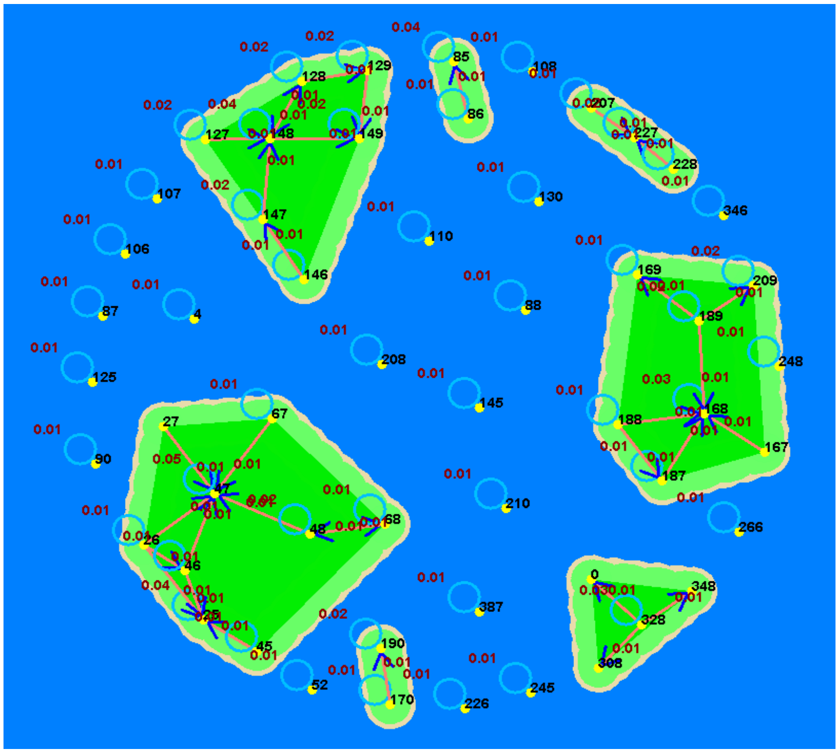


Figure C.9: October 2005 Type 1 Combination Patterns Probability Map

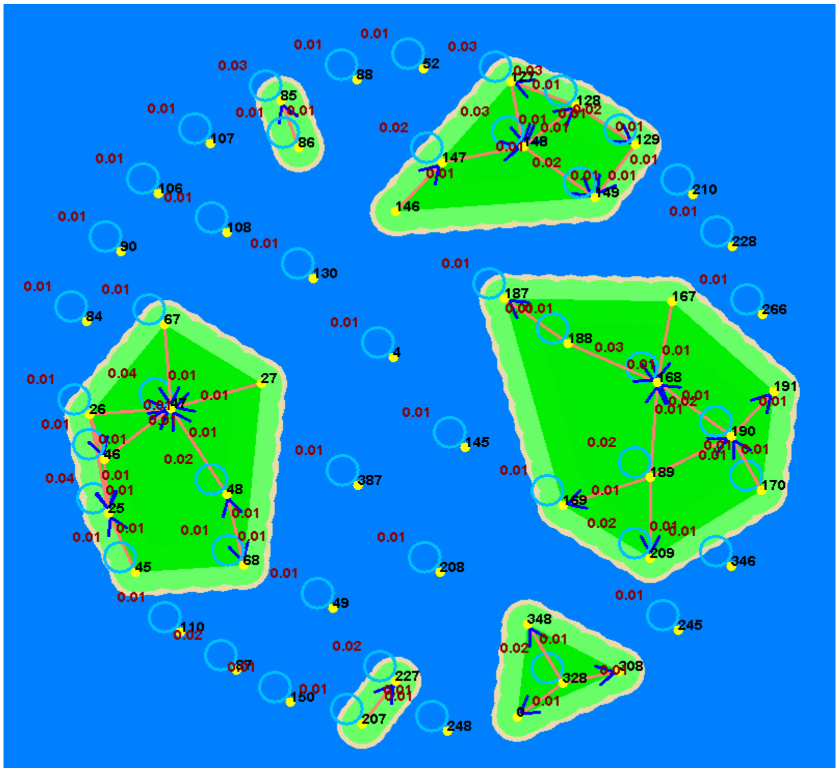


Figure C.10: November 2005 Type 1 Combination Patterns Probability Map

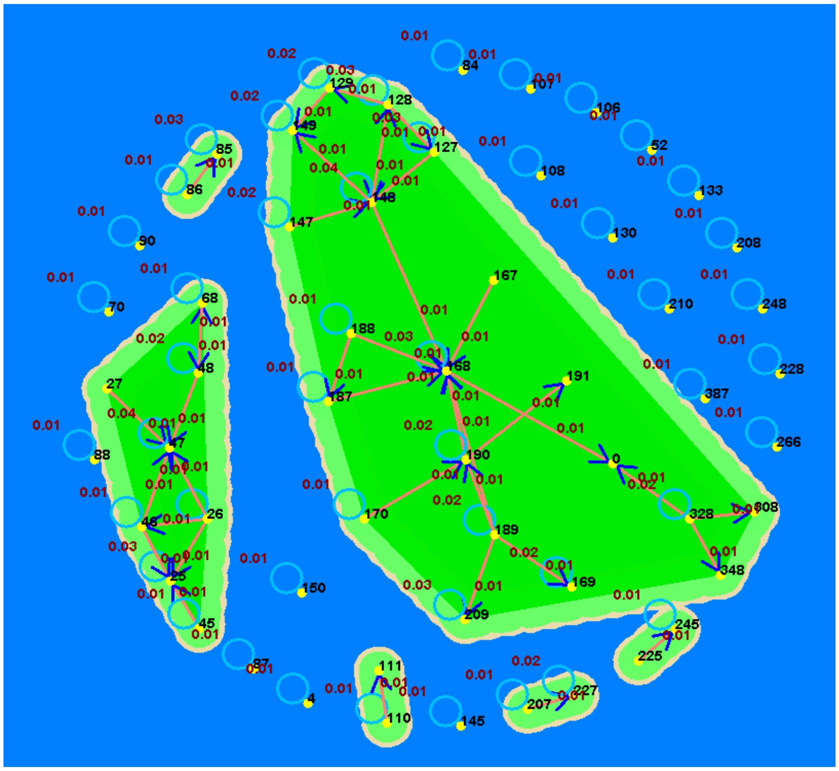


Figure C.11: December 2005 Type 1 Combination Patterns Probability Map

Appendix D

Probability Maps for CTS Type 1 Combination Patterns between February and December 2006

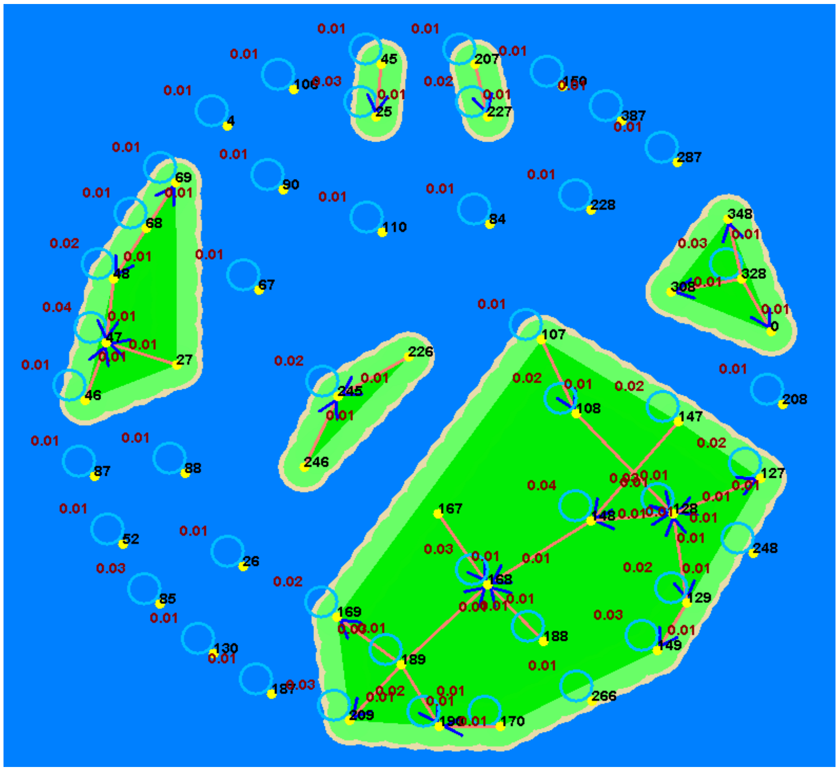


Figure D.1: February 2006 Type 1 Combination Patterns Probability Map

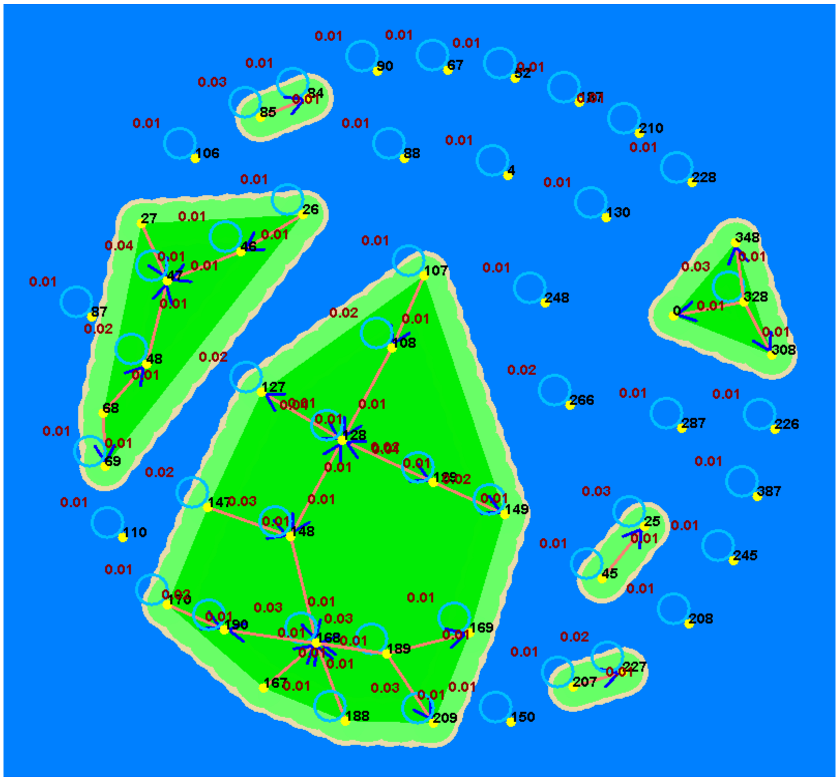


Figure D.2: March 2006 Type 1 Combination Patterns Probability Map

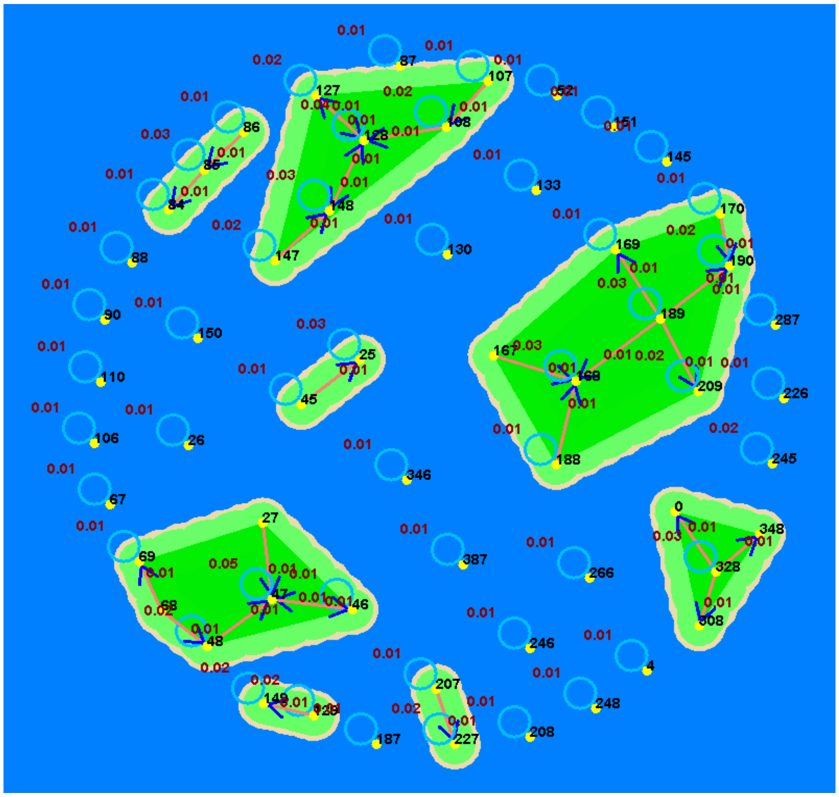


Figure D.3: April 2006 Type 1 Combination Patterns Probability Map

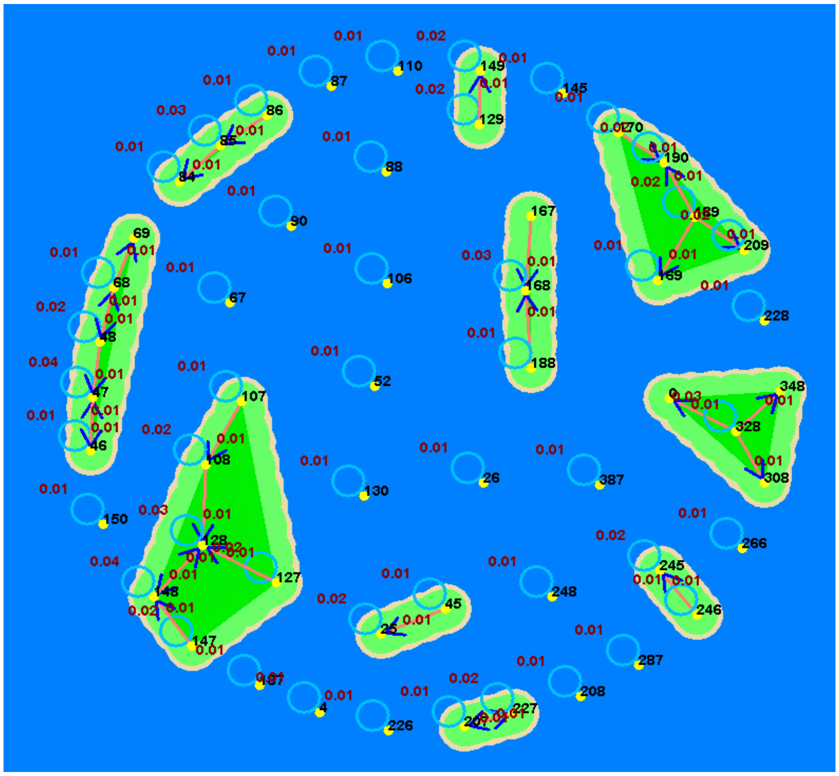


Figure D.4: May 2006 Type 1 Combination Patterns Probability Map

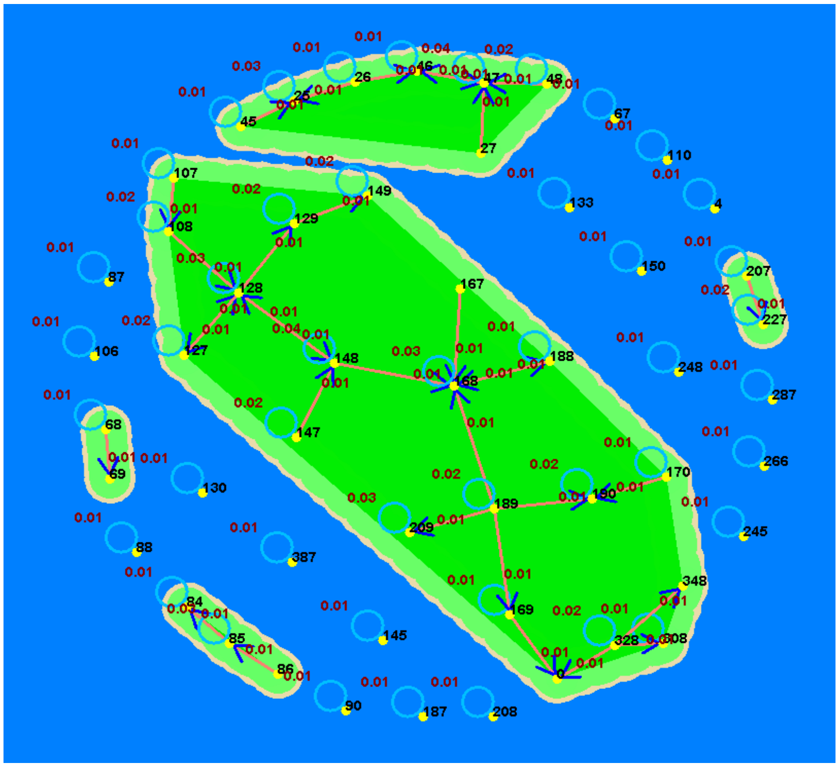


Figure D.5: June 2006 Type 1 Combination Patterns Probability Map

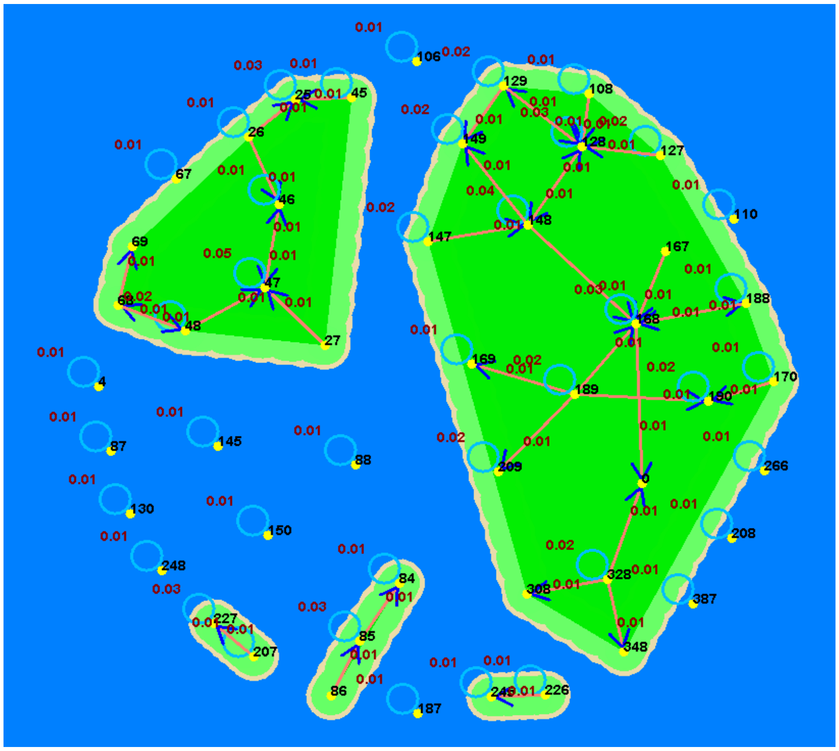


Figure D.6: July 2006 Type 1 Combination Patterns Probability Map

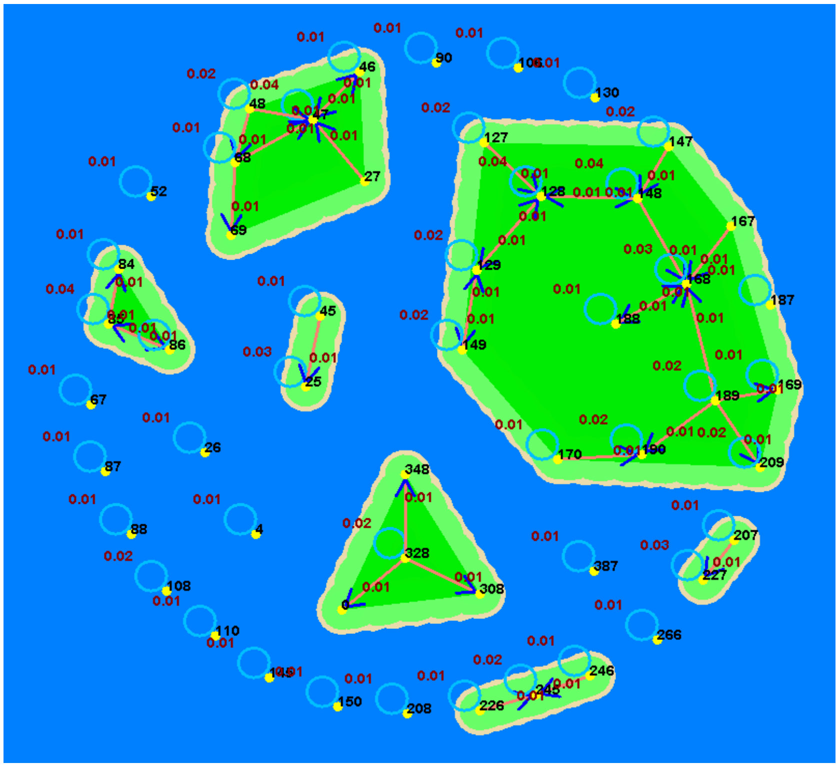


Figure D.7: August 2006 Type 1 Combination Patterns Probability Map

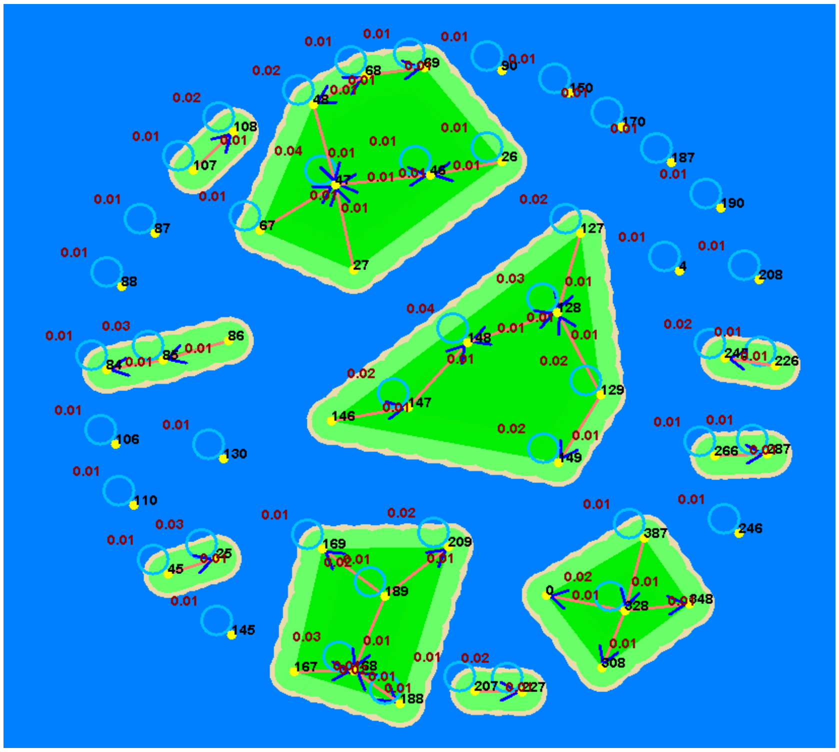


Figure D.8: September 2006 Type 1 Combination Patterns Probability Map

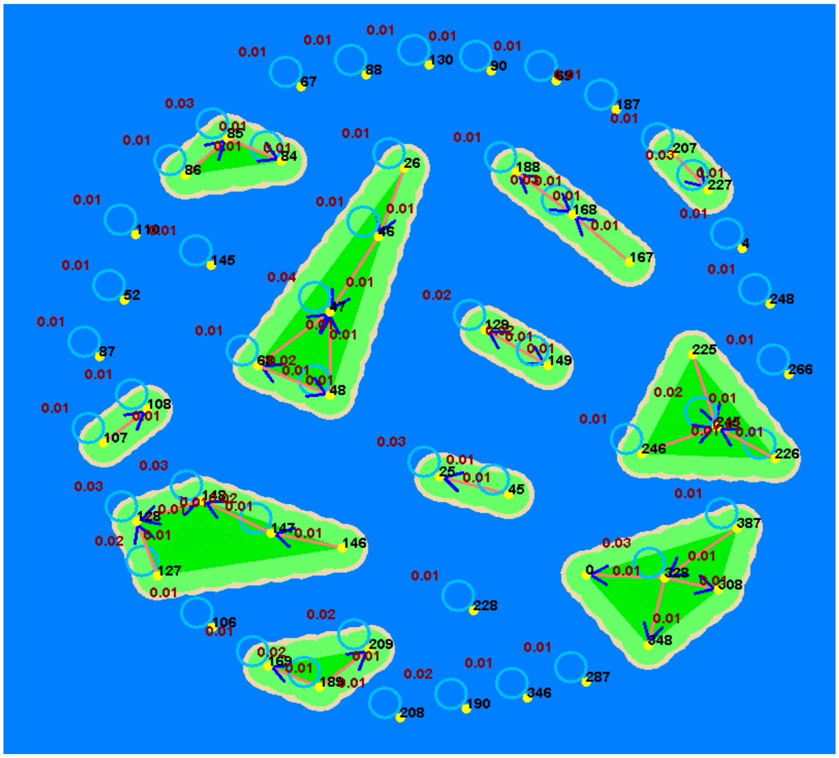


Figure D.9: October 2006 Type 1 Combination Patterns Probability Map

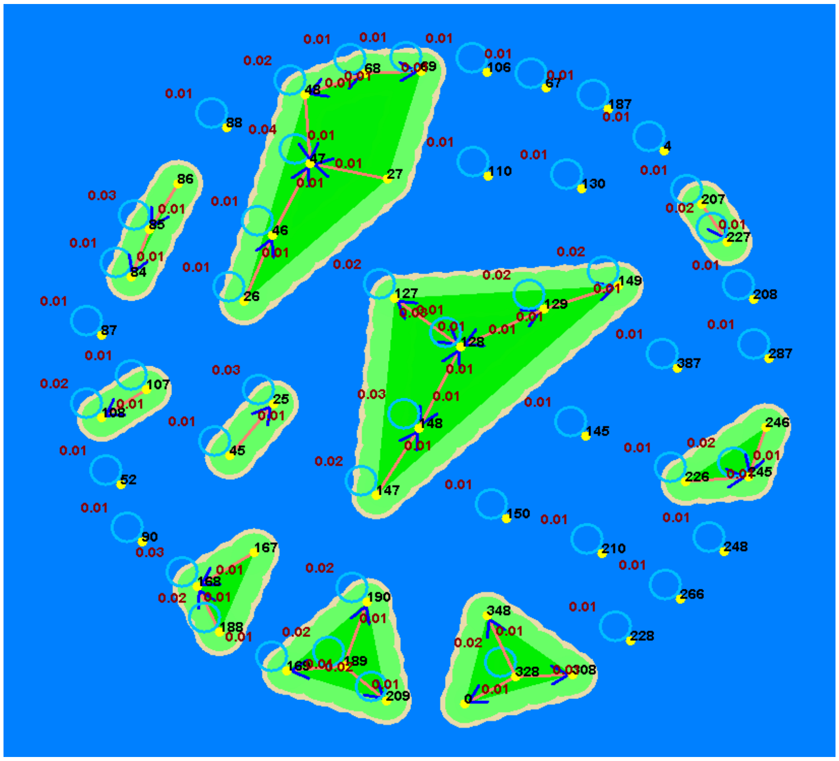


Figure D.10: November 2006 Type 1 Combination Patterns Probability Map

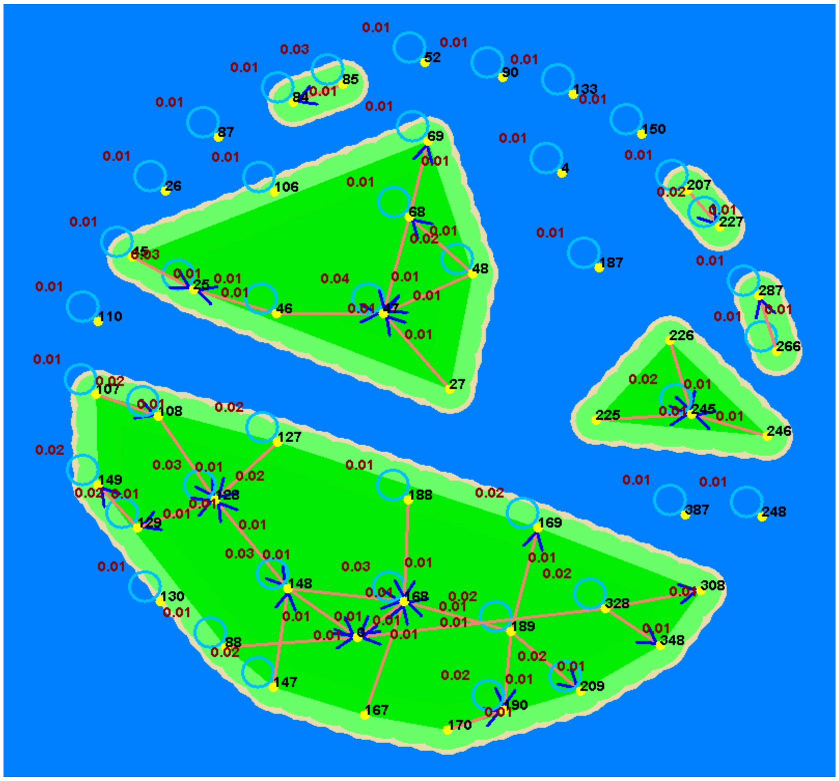


Figure D.11: December 2006 Type 1 Combination Patterns Probability Map

Appendix E

Probability Maps for CTS Type 2 Combination Patterns between June and December 2003

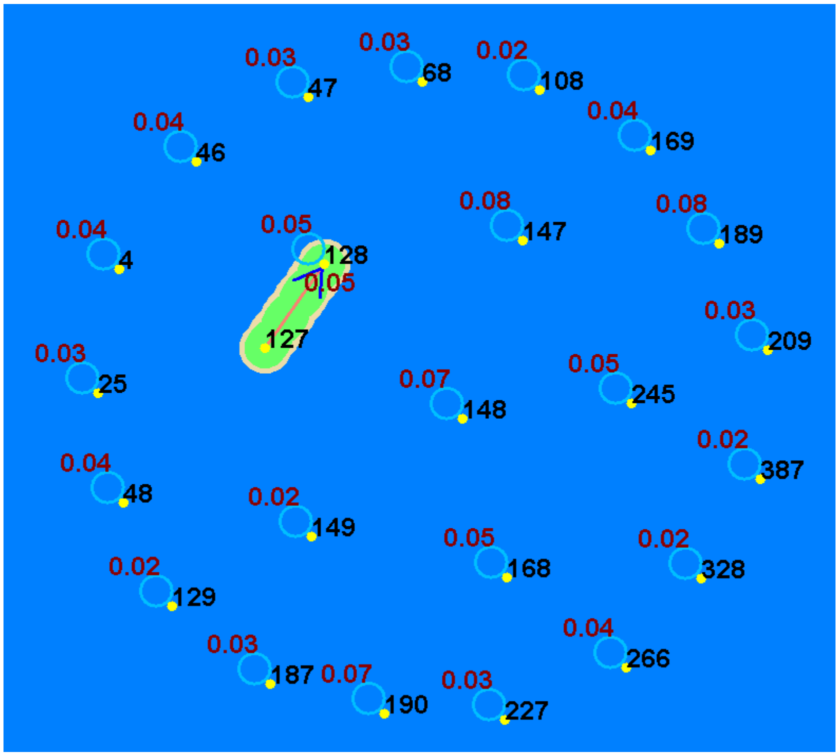


Figure E.1: June 2003 Type 2 Combination Patterns Probability Map

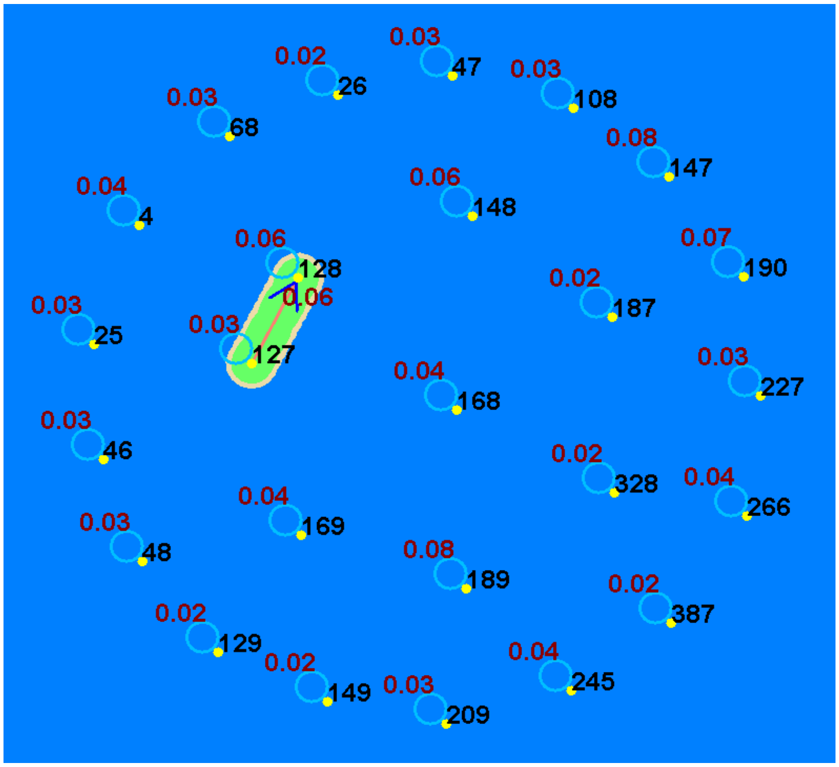


Figure E.2: July 2003 Type 2 Combination Patterns Probability Map

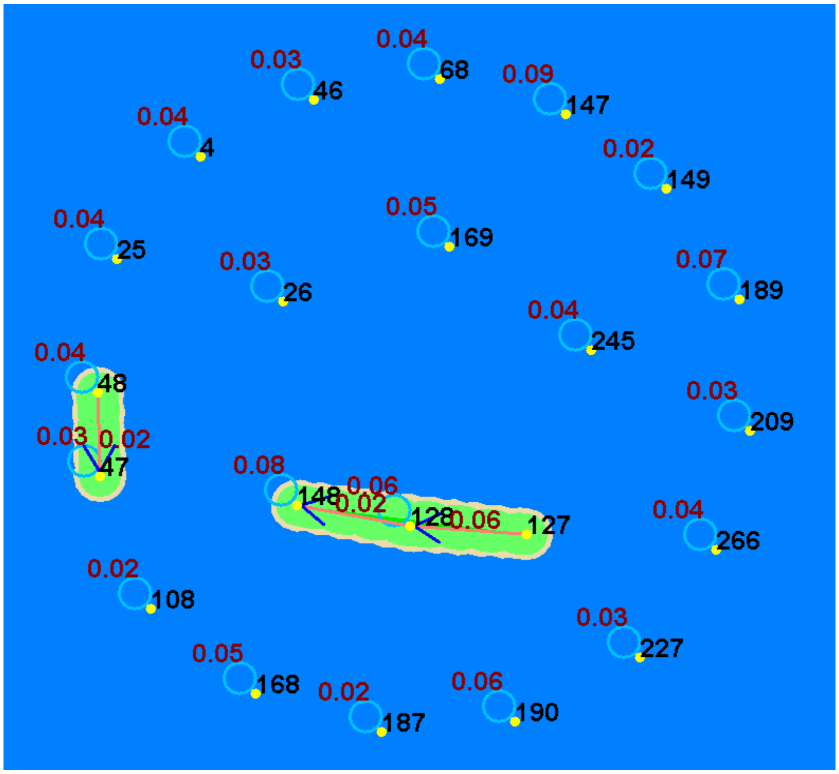


Figure E.3: August 2003 Type 2 Combination Patterns Probability Map

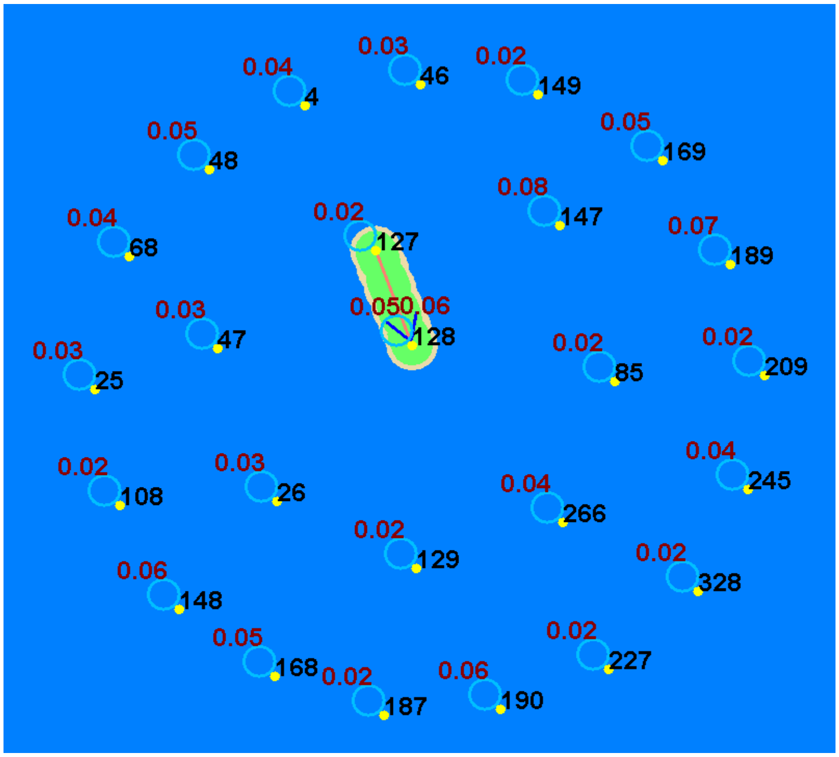


Figure E.4: September 2003 Type 2 Combination Patterns Probability Map

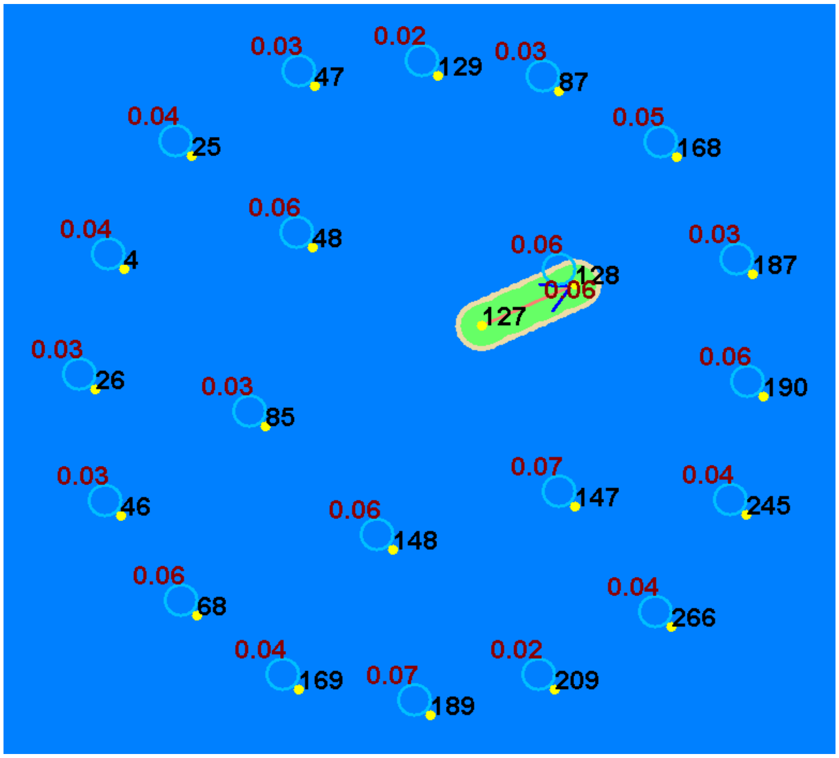


Figure E.5: October 2003 Type 2 Combination Patterns Probability Map

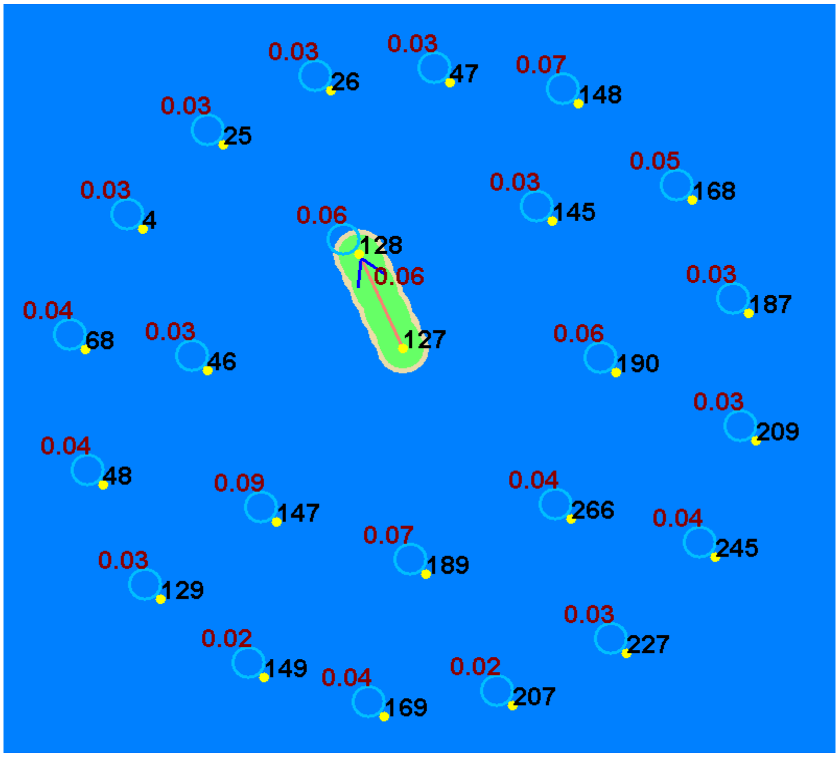


Figure E.6: November 2003 Type 2 Combination Patterns Probability Map

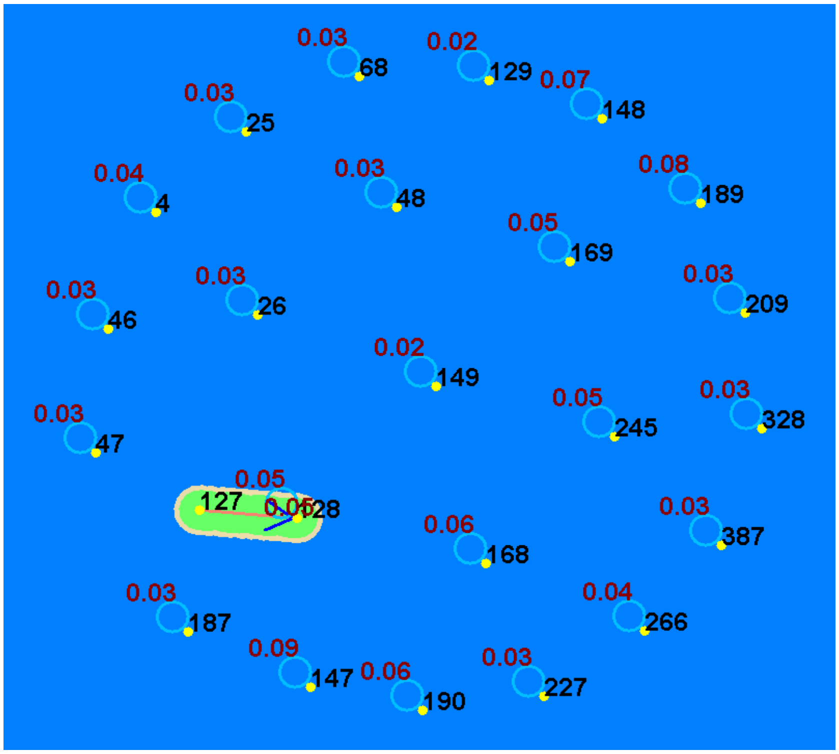


Figure E.7: December 2003 Type 2 Combination Patterns Probability Map

Appendix F

Probability Maps for CTS Type 2 Combination Patterns between February and December 2004

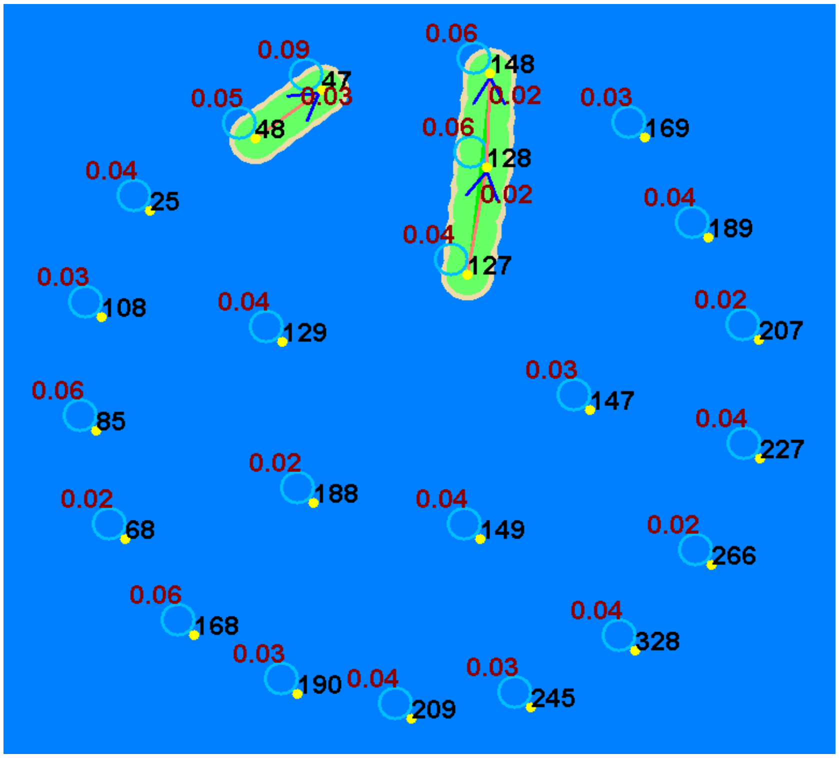


Figure F.1: February 2004 Type 2 Combination Patterns Probability Map

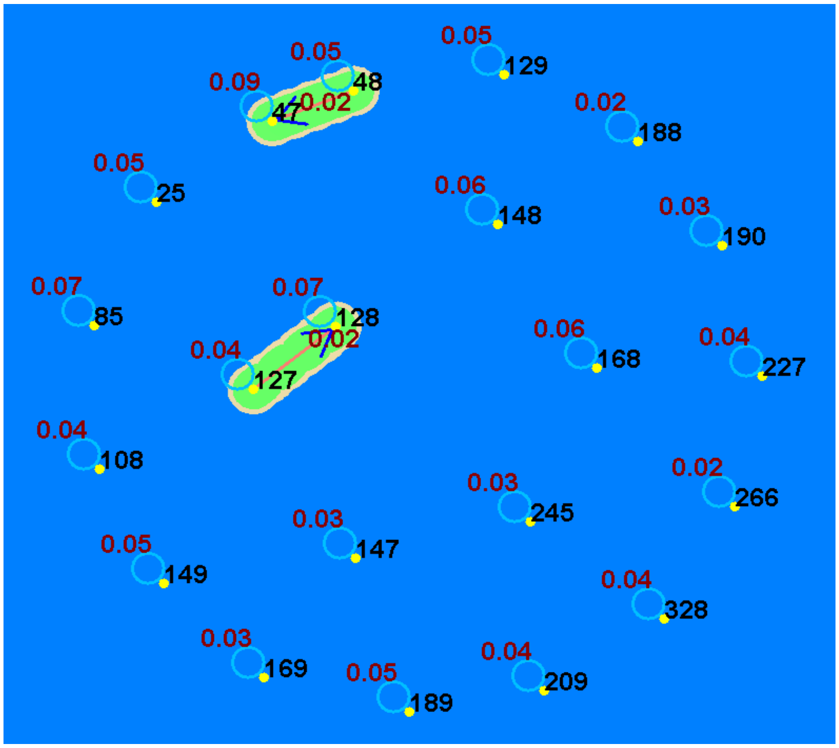


Figure F.2: March 2004 Type 2 Combination Patterns Probability Map

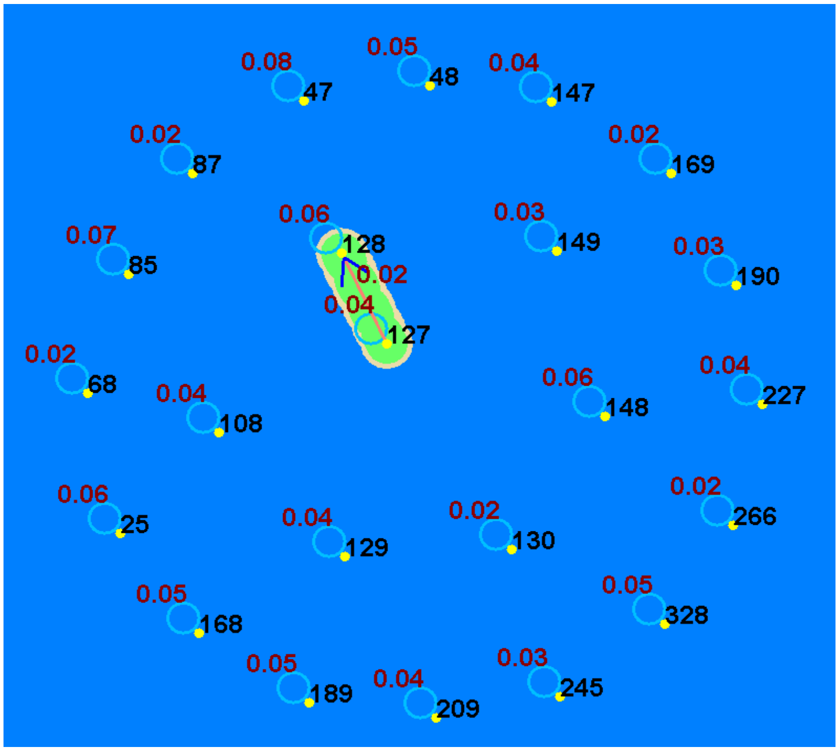


Figure F.3: April 2004 Type 2 Combination Patterns Probability Map

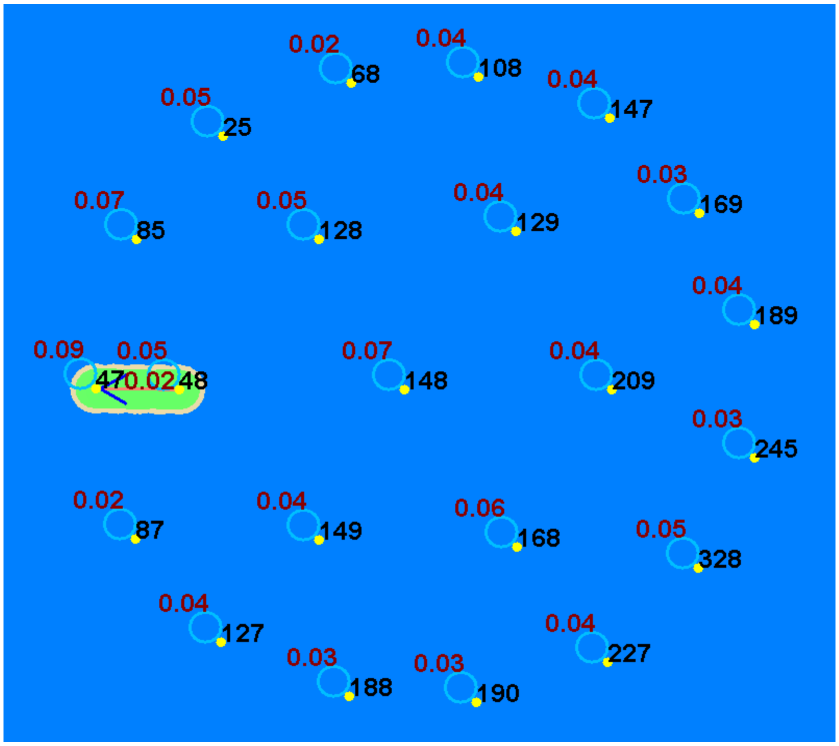


Figure F.4: May 2004 Type 2 Combination Patterns Probability Map

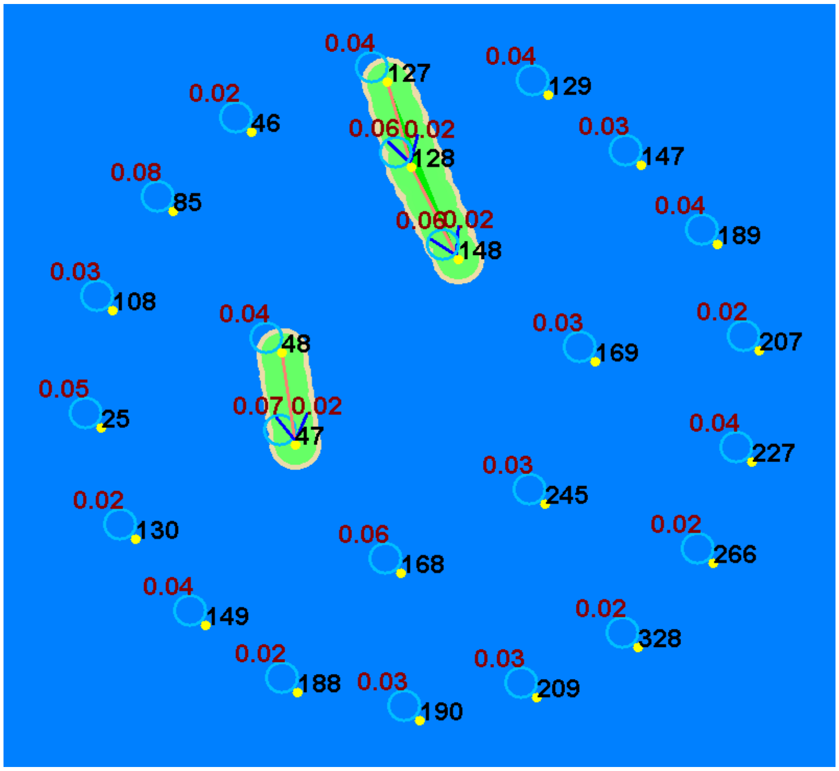


Figure F.5: June 2004 Type 2 Combination Patterns Probability Map

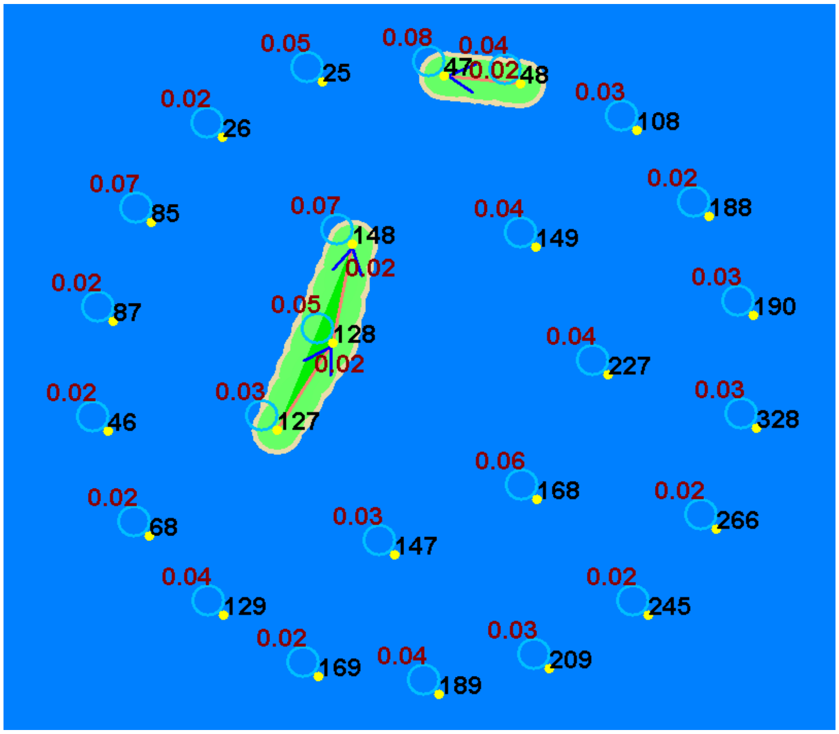


Figure F.6: July 2004 Type 2 Combination Patterns Probability Map

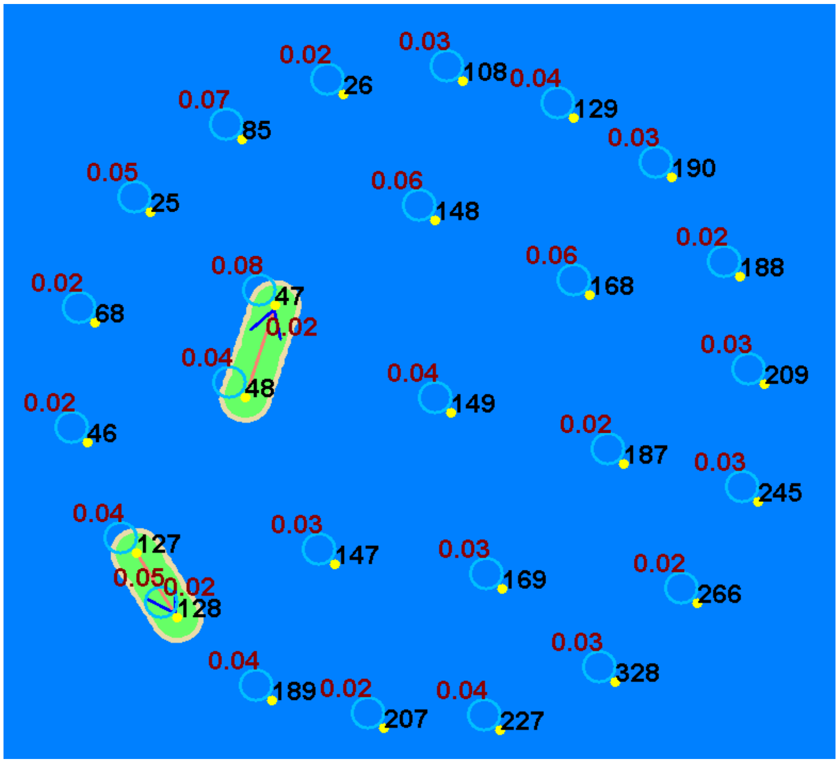


Figure F.7: August 2004 Type 2 Combination Patterns Probability Map

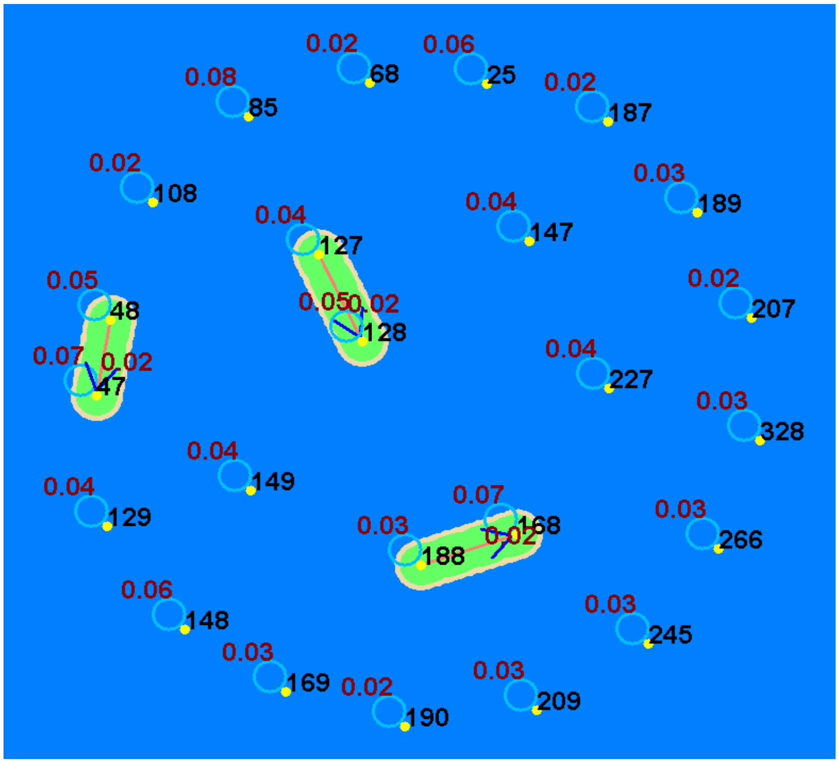


Figure F.8: September 2004 Type 2 Combination Patterns Probability Map

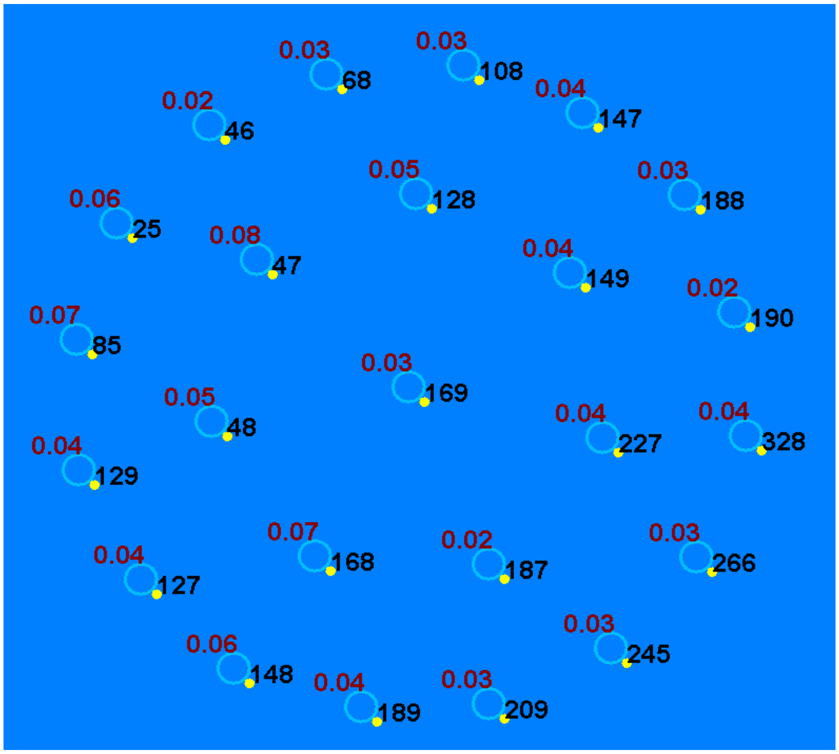


Figure F.9: October 2004 Type 2 Combination Patterns Probability Map

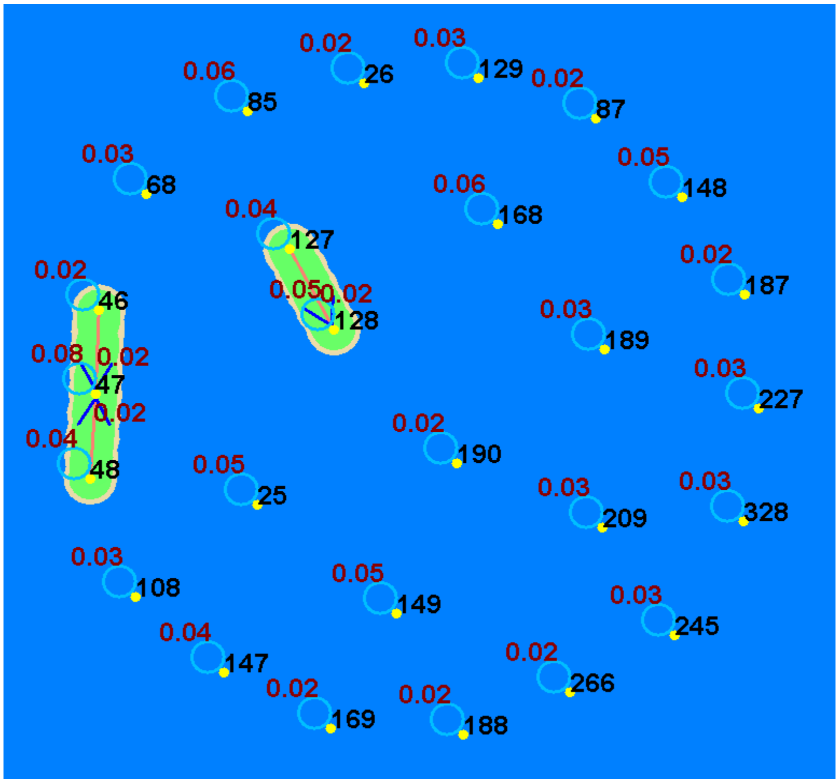


Figure F.10: November 2004 Type 2 Combination Patterns Probability Map

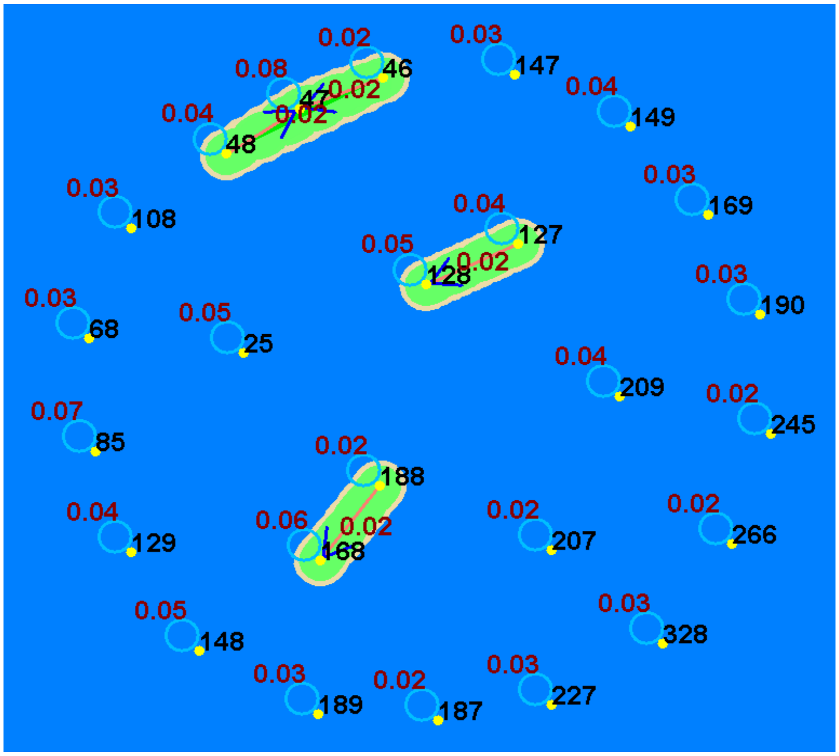


Figure F.11: December 2004 Type 2 Combination Patterns Probability Map

Appendix G

Probability Maps for CTS Type 2 Combination Patterns between February and December 2005

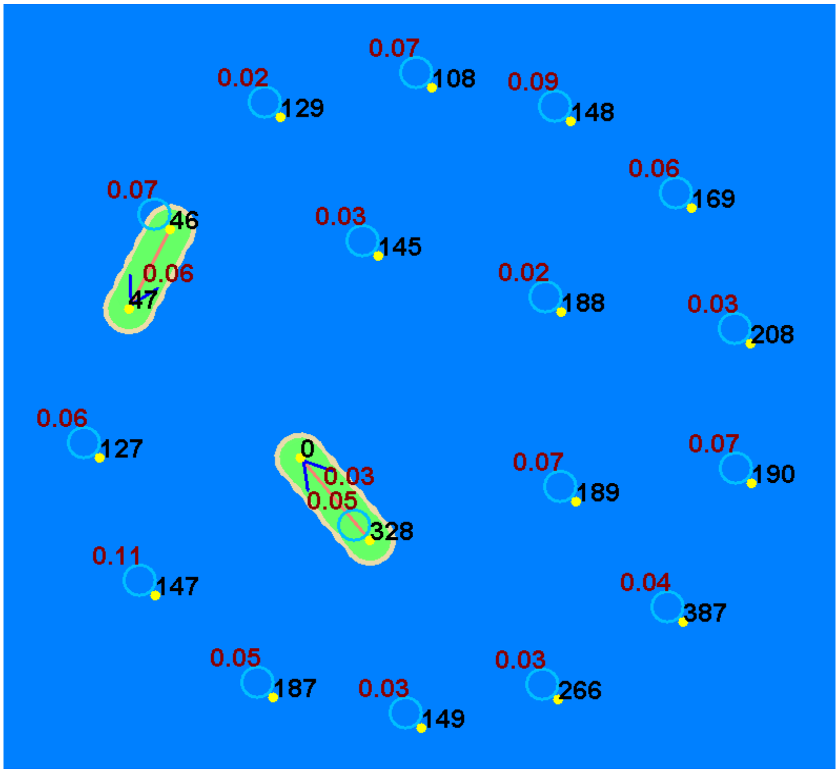


Figure G.1: February 2005 Type 2 Combination Patterns Probability Map

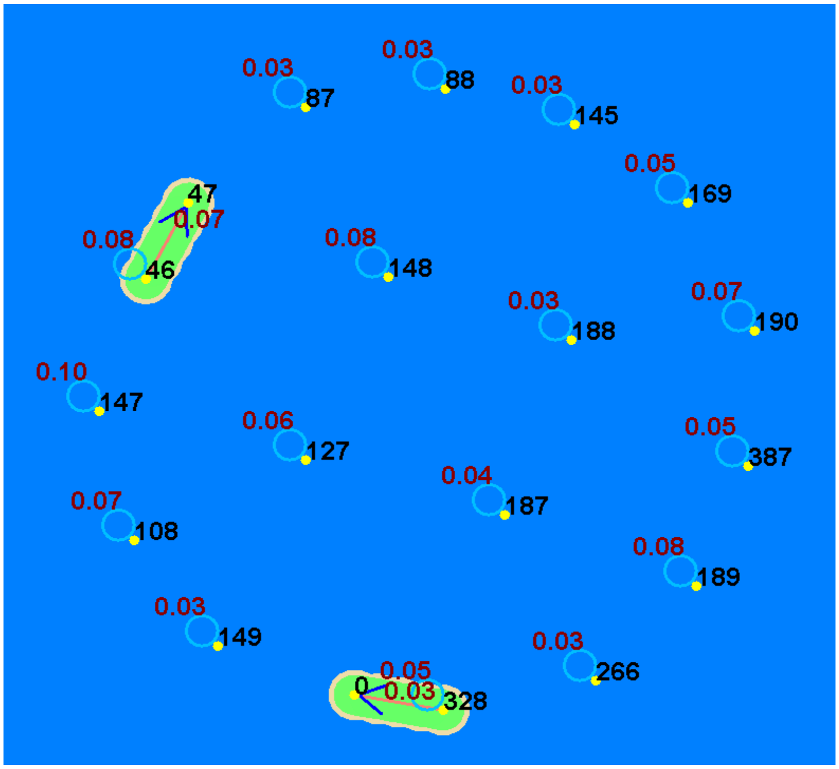


Figure G.2: March 2005 Type 2 Combination Patterns Probability Map

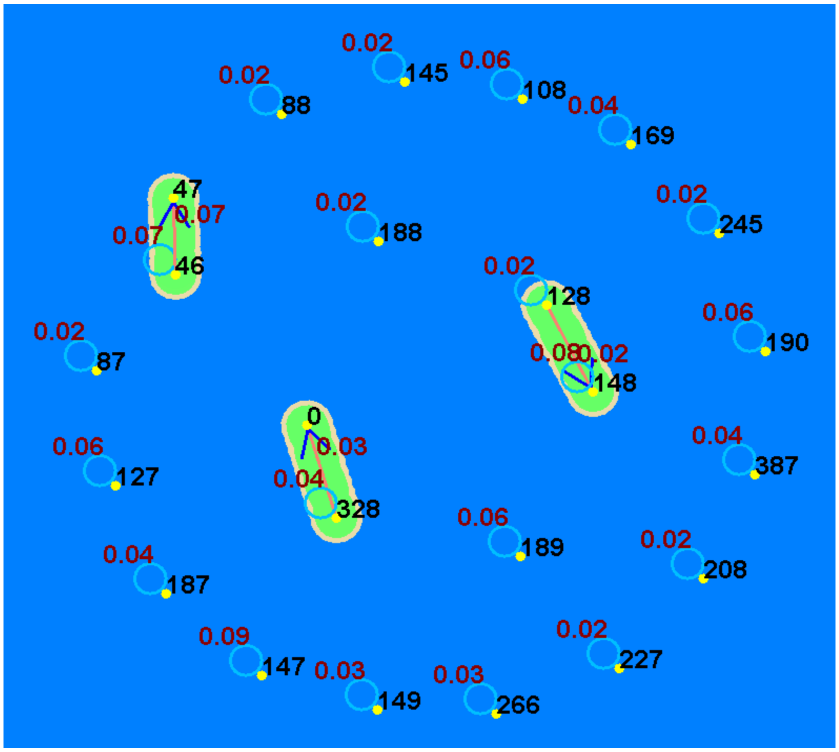


Figure G.3: April 2005 Type 2 Combination Patterns Probability Map

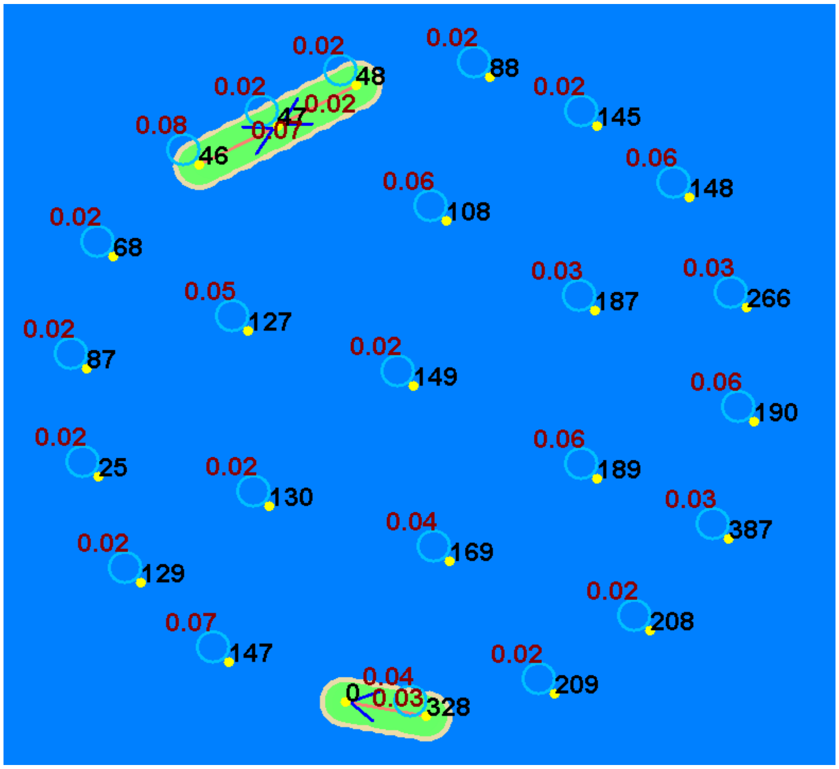


Figure G.4: May 2005 Type 2 Combination Patterns Probability Map

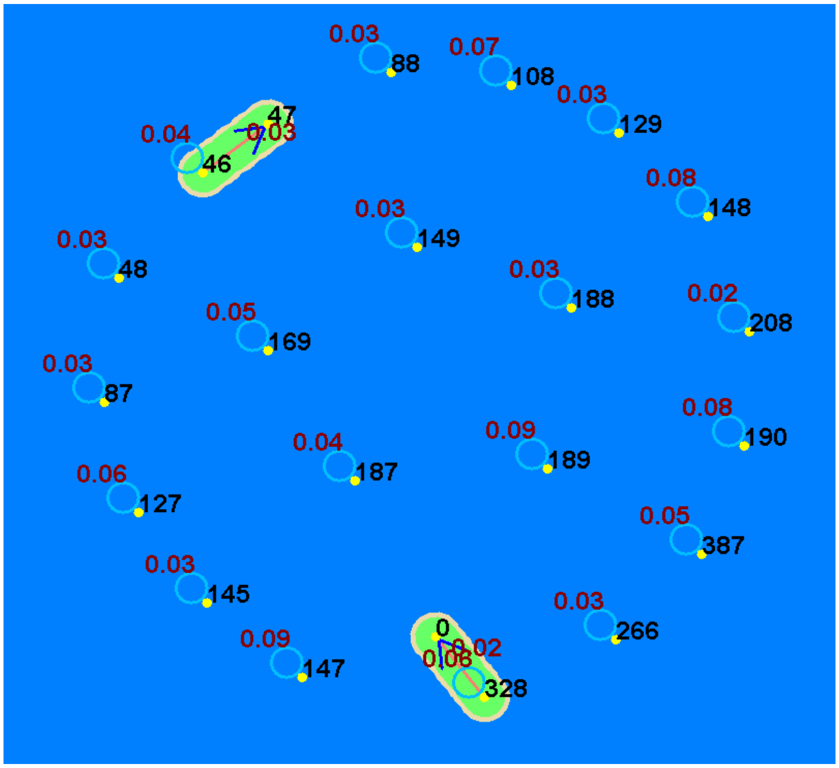


Figure G.5: June 2005 Type 2 Combination Patterns Probability Map

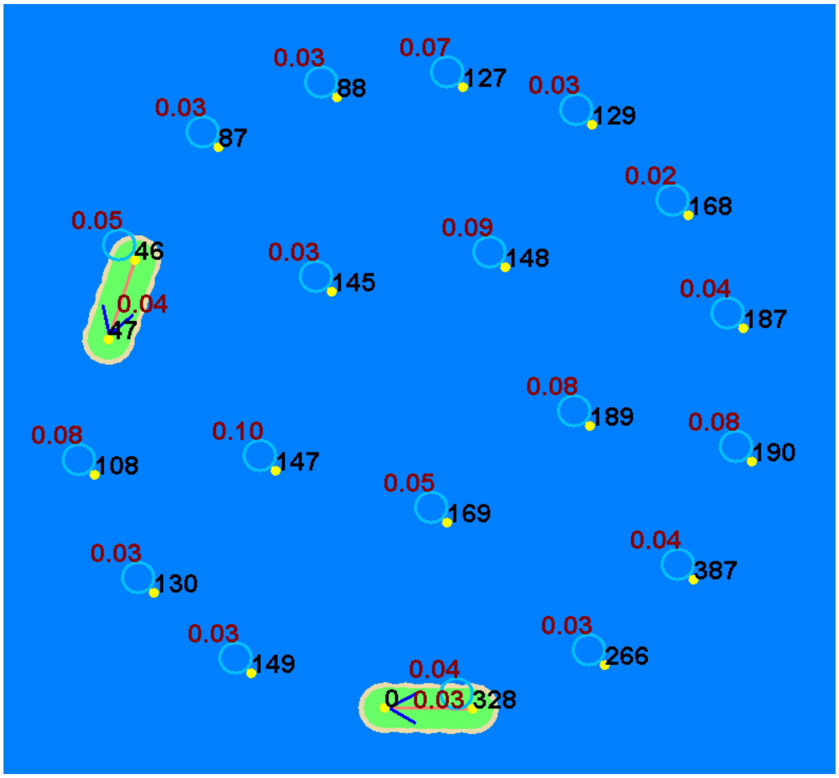


Figure G.6: July 2005 Type 2 Combination Patterns Probability Map

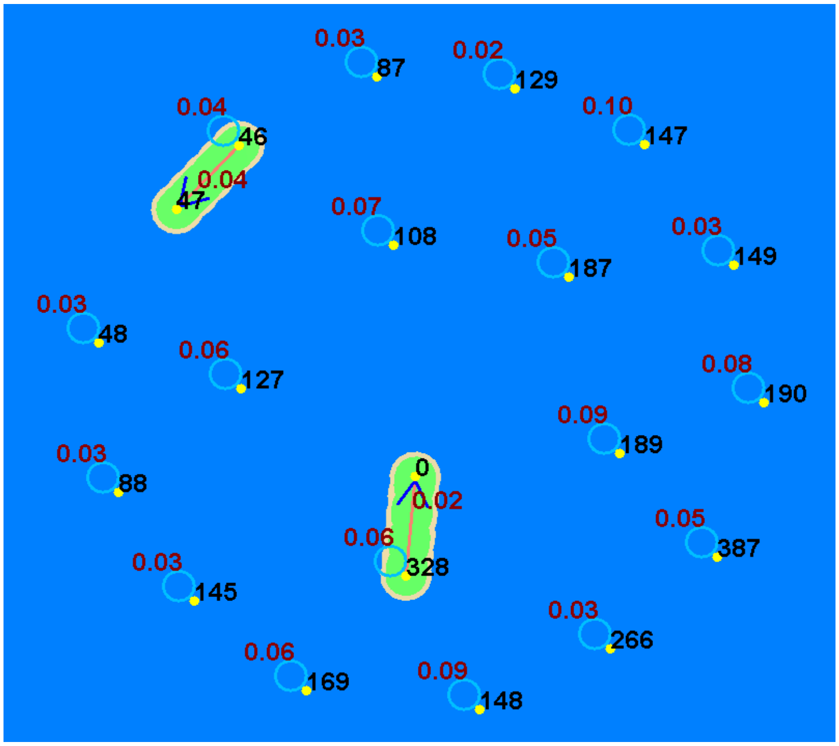


Figure G.7: August 2005 Type 2 Combination Patterns Probability Map

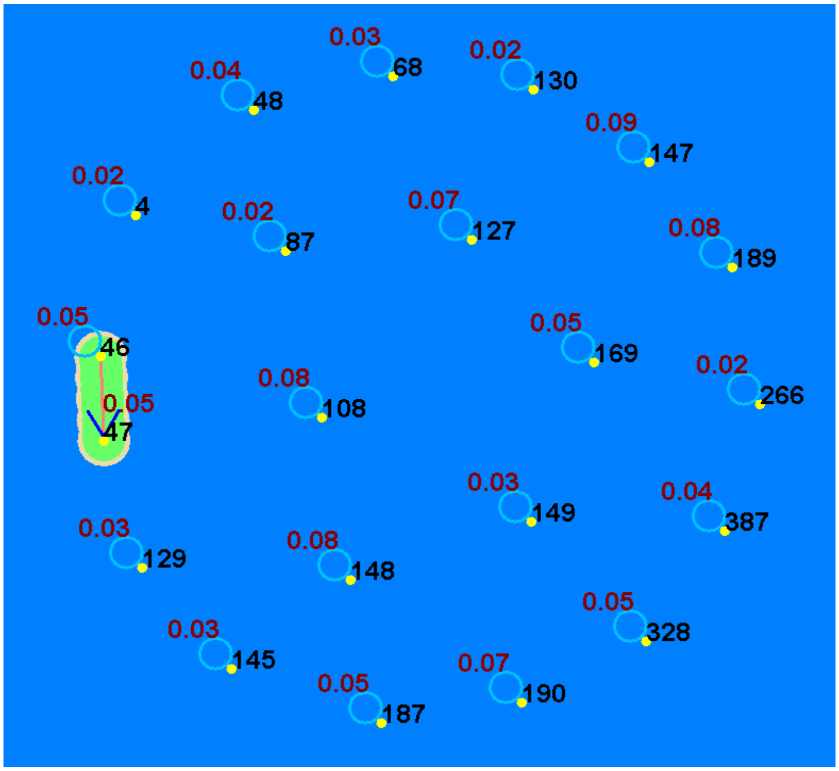


Figure G.8: September 2005 Type 2 Combination Patterns Probability Map

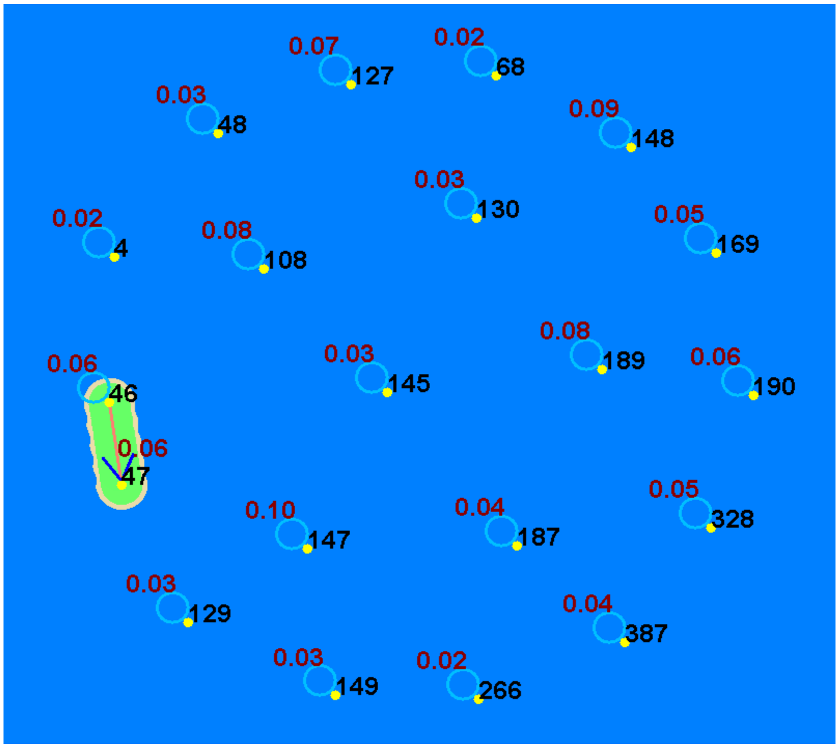


Figure G.9: October 2005 Type 2 Combination Patterns Probability Map

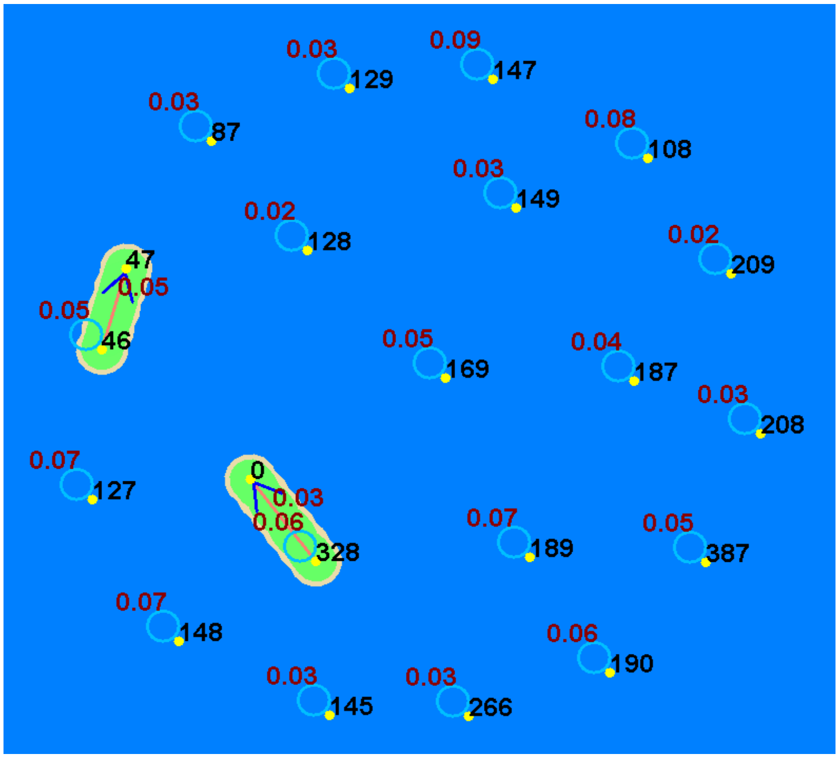


Figure G.10: November 2005 Type 2 Combination Patterns Probability Map

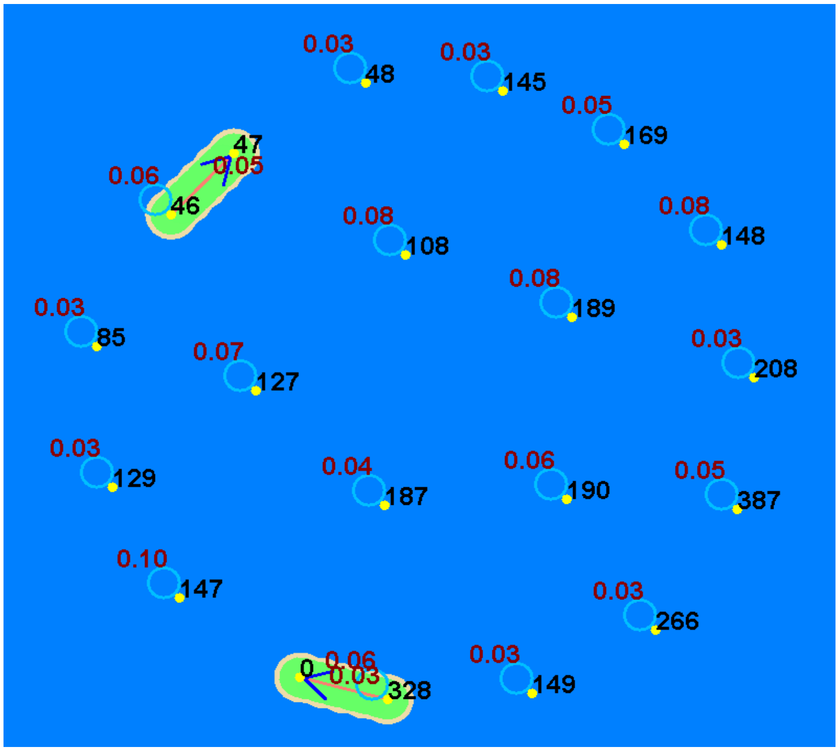


Figure G.11: December 2005 Type 2 Combination Patterns Probability Map

Appendix H

Probability Maps for CTS Type 2 Combination Patterns between February and December 2006

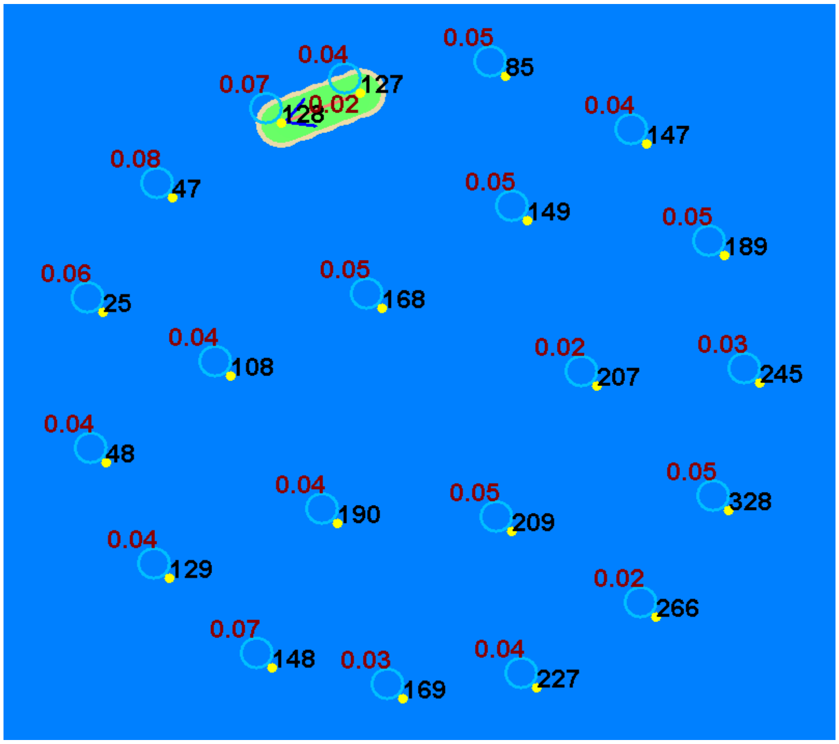


Figure H.1: February 2006 Type 2 Combination Patterns Probability Map

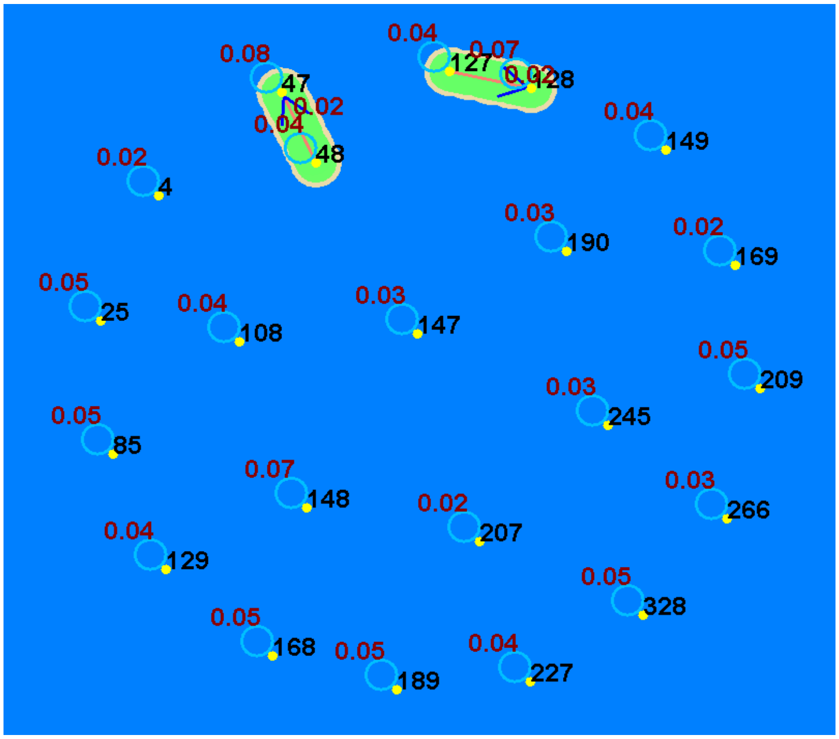


Figure H.2: March 2006 Type 2 Combination Patterns Probability Map

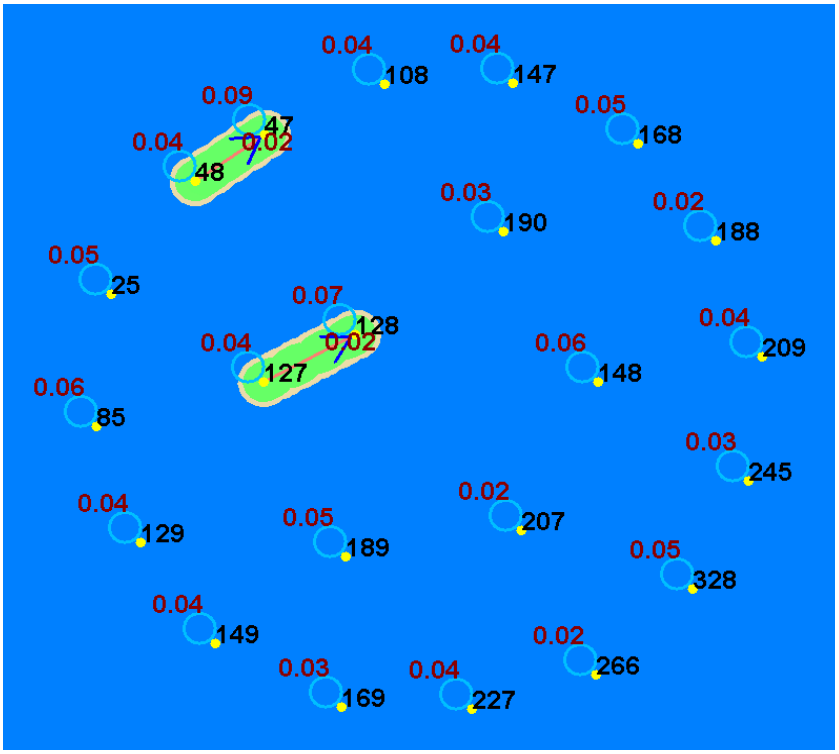


Figure H.3: April 2006 Type 2 Combination Patterns Probability Map

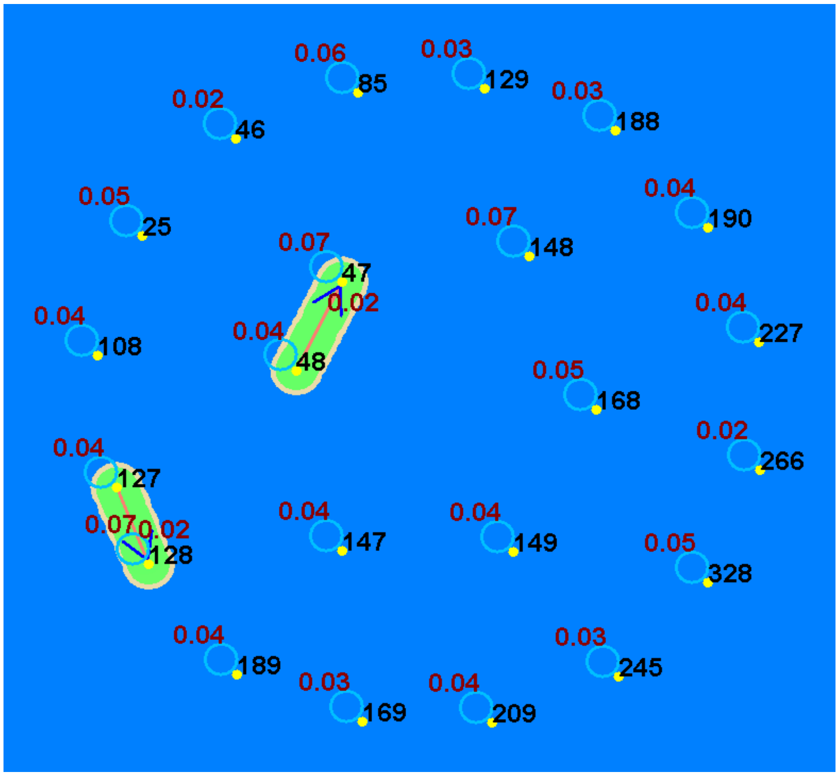


Figure H.4: May 2006 Type 2 Combination Patterns Probability Map

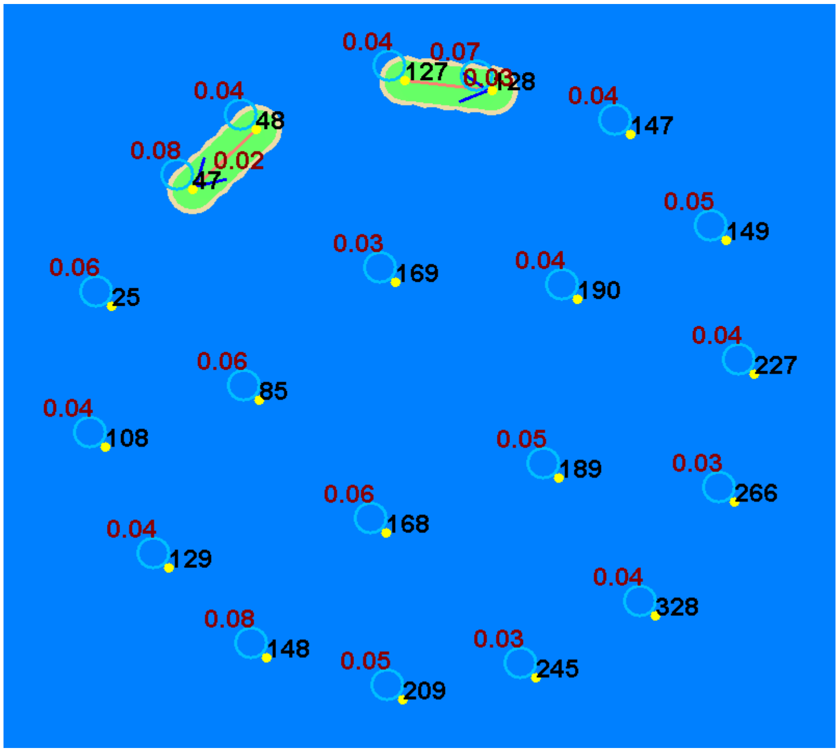


Figure H.5: June 2006 Type 2 Combination Patterns Probability Map

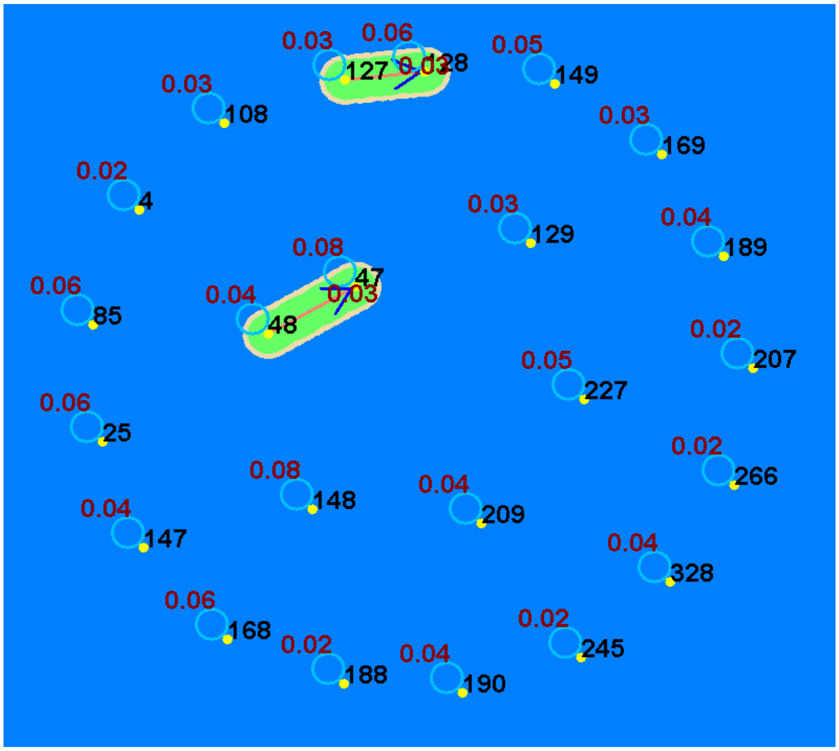


Figure H.6: July 2006 Type 2 Combination Patterns Probability Map

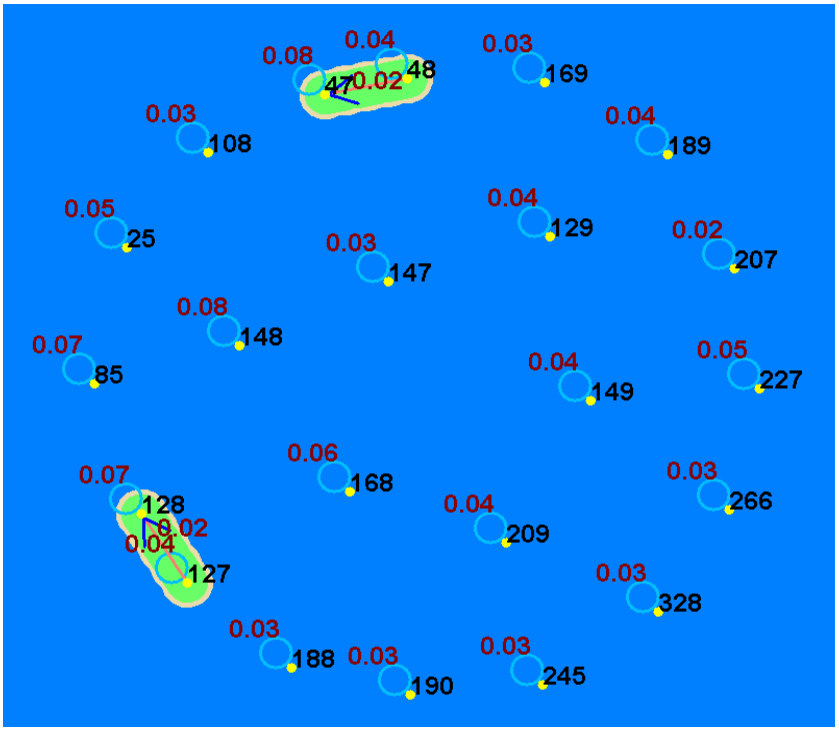


Figure H.7: August 2006 Type 2 Combination Patterns Probability Map

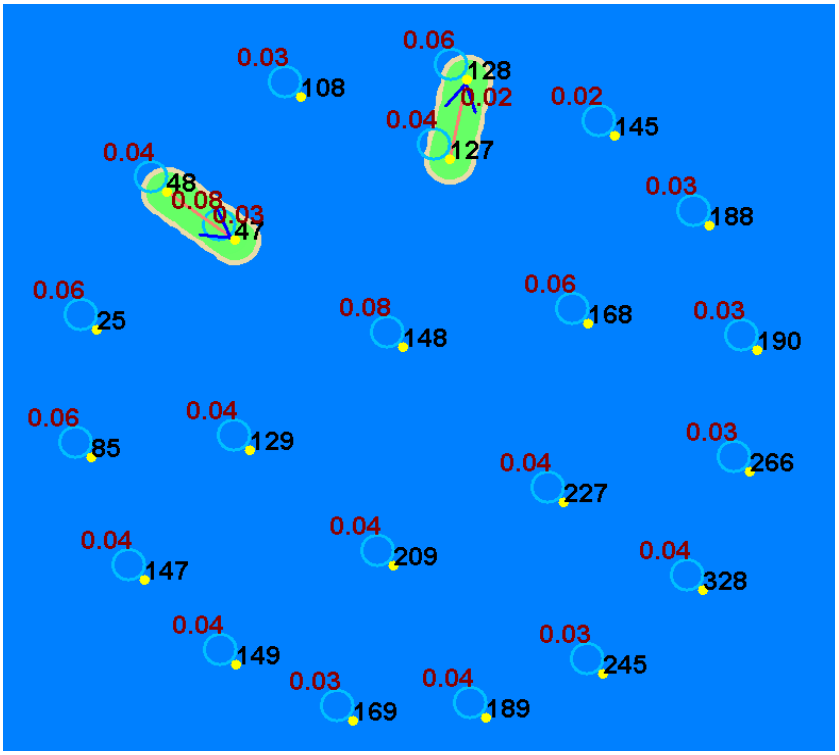


Figure H.8: September 2006 Type 2 Combination Patterns Probability Map

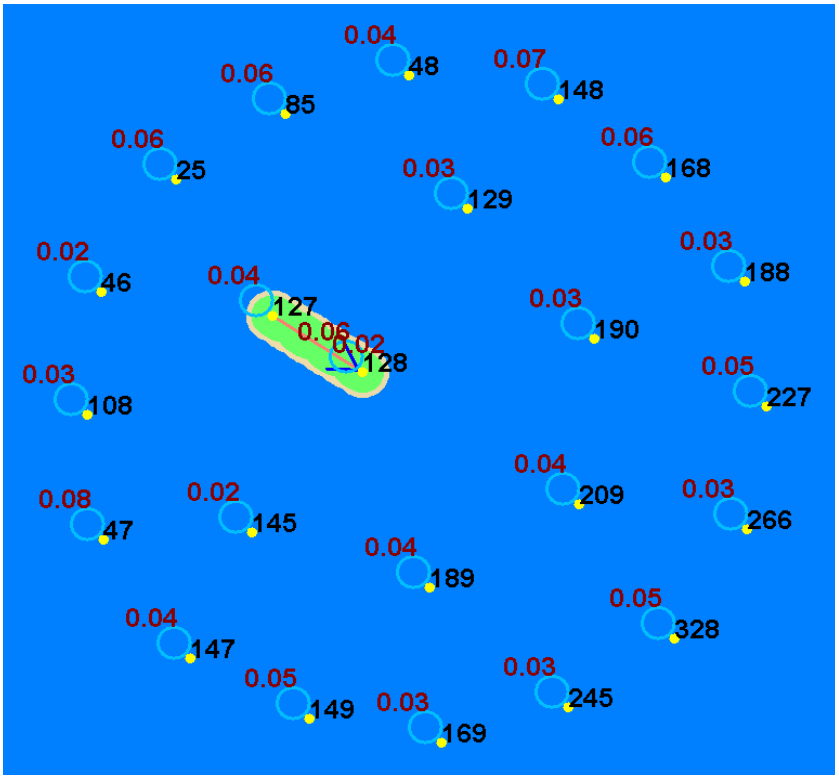


Figure H.9: October 2006 Type 2 Combination Patterns Probability Map

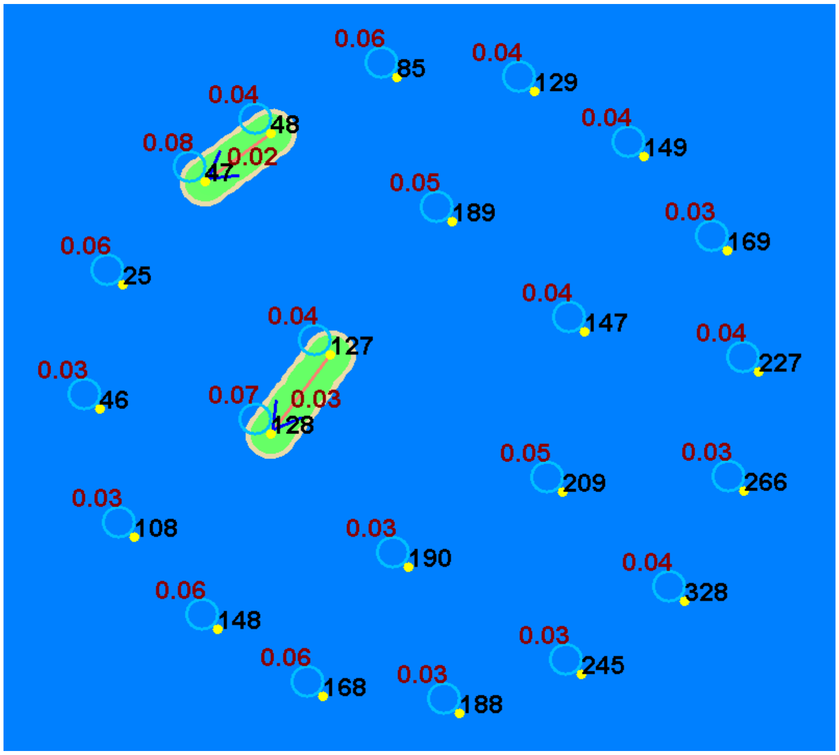


Figure H.10: November 2006 Type 2 Combination Patterns Probability Map

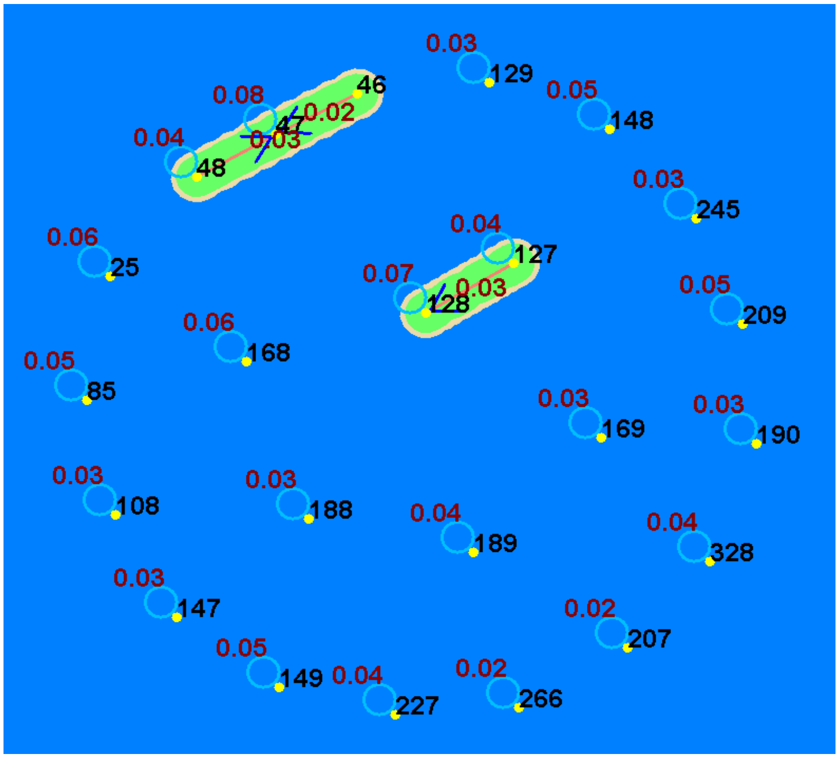


Figure H.11: December 2006 Type 2 Combination Patterns Probability Map

Bibliography

[1] Predictive modelling technology. http://www.predx.com_/docs_/PredxModelingTechnology.pdf, 2004. [Online; accessed 11-Nov-2011].

[2] Earth friends, a social network visualization. http://www.multigesture.net/2010/12/31/earth-friends-a-social-network-visualization/, 2010. [Online; accessed 19-Feb-2012].

[3] Overcoming data mining challenges. http://www.rexeranalytics.com/Overcoming_Challenges.html, 2010. [Online; accessed 11-Nov-2011].

[4] D. Abbott, I. Matkovsky, and J. Elder. An evaluation of high-end data mining tools for fraud detection. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 12–14, 1998.

[5] G. Adomavicius and J. Bockstedt. C-trend: Temporal cluster graphs for identifying and visualizing trends in multiattribute transactional data. *IEEE Transaction on Knowledge and Data Engineering*, 20:721–735, June 2008.

[6] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.

[7] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[8] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, 1995.

[9] S. Ahmed, F. Coenen, and P. Leng. A tree partitioning method for memory management in association rule mining. In *DaWaK*, pages 331–340, 2004.

[10] R. Alves, D. Rodriguez-Baena, and J. Aguilar-Ruiz. Gene association analysis: A survey of frequent pattern mining from gene expression data. *Briefings in Bioinformatics*, 11(2):210–224, 2010.

[11] M. Angermeyer and H. Matschinger. Causal beliefs and attitudes to people with schizophrenia: Trend analysis based on data from two population surveys in germany. *The British Journal of Psychiatry*, 186:331–334, 2005.

[12] C. Antunes and A. Oliveira. Temporal data mining: An overview. In *Proceedings ACM SIGKDD Workshop Data Mining*, pages 1–13, 2001.

[13] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, pages 61–70, 2010.

[14] P. Bala. Retail inventory management with purchase dependencies. *International Journal of Engineering Letters*, 16(4):545–549, 2008.

[15] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *Proceedings of the 2006 International Workshop on Mining Software Repositories*, pages 137–143, 2006.

[16] E. Bloedorn, A. Christiansen, W. Hill, C. Skorupka, L. Talbot, and J. Tivel. Data mining for network intrusion detection: How to get started. Technical report, The MITRE Corporation, 2001.

[17] J. Bobadilla, F. Serradilla, and J. Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. *Journal of Knowledge Based System*, 23(6):520–528, 2010.

[18] C. Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, pages 1–5. ACM Press, 2005.

[19] R. Brause, T. Langsdorf, and M. Hepp. Neural data mining for credit card fraud detection. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, 1999.

[20] P. Brockwell and R. Davis. *Time Series:Theory and Methods*. Springer, 2001.

[21] S. Buckland, A. Magurran, R. Green, and R. Fewster. Monitoring change in biodiversity through composite indices. *Philosophical Transaction of Royal Society Biological Sciences*, 360(1454):243–254, 2005.

[22] J. Caldwell. The box-jenkins forecasting technique. Technical report, Foundation, 2006.

[23] M. Ceci and A. Appice. Spatial associative classification: Propositional vs structural approach. *Journal of Intelligent Information System*, 27(3):191–213, 2006.

[24] B. Chen, Q. Zhao, B. Sun, and P. Mitra. Predicting blogging behavior using temporal and social networks. In *Proceedings of 2007 IEEE International Conference on Data Mining*, pages 439–444, 2007.

[25] C. Chen. Citespace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57:359–377, 2006.

[26] R. Christley, G. Pinchbeck, R. Bowers, D. Clancy, N. French, R. Bennett, and J. Turner. Practice of epidemiology infection in social networks: Using network analysis to identify high-risk individuals. *American Journal of Epidemiology*, 162(10):1024–1031, 2005.

[27] H. Chung and P. Gray. Special section: Data mining. *Journal of Management Information Systems*, 16:11–16, June 1999.

[28] K. Cios, W. Pedrycz, R. Swiniarski, and L. Kurgan. *Data Mining: A Knowledge Discovery Approach*. Springer-Verlag, 2007.

[29] F. Coenen, G. Goulbourne, and P. Leng. Computing association rules using partial totals. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 54–66, 2001.

[30] F. Coenen, P. Leng, and S. Ahmed. Data structures for association rule mining: T-trees and p-trees. *IEEE Transactions on Data and Knowledge Engineering*, 16(6):774–778, 2004.

[31] M. Cottrell and P Rousset. A powerful tool for analyzing and representing multidimensional quantitative and qualitative data. In *Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Biological and Artificial Computation: From Neuroscience to Technology*, pages 861–871, 1997.

[32] J. Davis. Understanding and decreasing aversive behavior in online social contexts. Technical report, American Association for Artificial Intelligence, 2002.

[33] E. de Graaf, J. Kok, and W. Kosters. Clustering co-occurrence of maximal frequent patterns in streams. *Computing Research Repository*, abs/0705.0588, 2007.

[34] D. Delen, G. Walker, and A. Kadam. Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2):113–127, 2005.

[35] Denny, G. Williams, and P. Christen. Visualizing temporal cluster changes using relative density self-organizing maps. *Knowledge and Information Systems*, 25(2):281–302, 2010.

[36] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan. Data mining for network intrusion detection. In *Proceedings of the NSF Workshop on Next Generation Data Mining*, 2002.

[37] P. Domingos. Mining social networks for viral marketing. *IEEE Intelligent Systems*, 20(1):80–82, 2005.

[38] M. Dunham. *Data Mining: Introductory and Advanced Topics.* Pearson Education, 2003.

[39] M. Esteban-Parra, F. Rodrigo, and Y. Castro-Diez. Temperature trends and change points in the northern spanish plateau during the last 100 years. *International Journal of Climatology*, 15:1031–1042, 1995.

[40] M. Ester, H. Kriegel, and J. Sander. *Algorithms and Applications for Spatial Data Mining.* Geographic Data Mining and Knowledge Discovery. London: Taylor and Francis, 2001.

[41] U. Fayyad. Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications*, 11(5):20–25, 1996.

[42] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases: an overview. *AI Magazine*, 13(3):57–70, 1992.

[43] P. Gloor, J. Krauss, S. Nann, K. Fischbach, and D. Schoder. Web science 2.0: Identifying trends through semantic social network analysis. social science research network. *Social Science Research Network Working Paper Series*, 4:215–222, 2008.

[44] B. Goethals. Survey on frequent pattern mining. Technical report, HIIT Basic Research Unit, University of Helsinki, 2003.

[45] M. Gorawski and P. Jureczek. A proposal of spatio-temporal pattern queries. In *Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 587–593, 2010.

[46] U. Gursoy. Customer churn analysis in telecommunication sector. *Istanbul University Journal of the School of Business Administration*, 39 (1):35–49, 2010.

[47] E. Hadavandi, H. Shavandi, and A. Ghanbari. Integration of genetic ffuzzy systems and artificial neural networks for stock price forecasting. *Journal of Knowledge-Based Systems*, 23:800–808, December 2010.

[48] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *Proceedings International Conference on Data Engineering*, pages 106–115, 1999.

[49] J. Han and J. Gao. *Research Challenges for Data Mining in Science and Engineering. Next Generation of Data Minin*, chapter 1, pages 3–28. Chapman and Hall, 2009.

[50] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques 3rd Edition*. Morgan Kaufmann, 2011.

[51] R. Hanneman and M. Riddle. *Introduction to social network methods*. University of California, Riverside, Riverside, CA, 2005.

[52] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002. get.

[53] S. Hido, T. Idé, H. Kashima, H. Kubo, and H. Matsuzawa. Unsupervised change analysis using supervised learning. In *Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 148–159. Springer-Verlag, 2008.

[54] T. Honjo, K. Umeki, E. Lim, D. Wang, P. Yang, and H. Hsieh. Landscape visualization on google earth. In *Proceedings of the 2009 Plant Growth Modeling, Simulation, Visualization, and Applications*, pages 445–448. IEEE Computer Society, 2009.

[55] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communication of the ACM*, 39(11):58–64, 1996.

[56] R. Ivancsy and I. Vajk. Frequent pattern mining in web log data. *Technology*, 3 (1):77–90, 2006.

[57] Han J and Kamber M. *Data Mining: Concepts and Techniques 2nd edition*. The Morgan Kaufmann Publishers, 2006.

[58] Kaye J., M. Melero-Montes, and Jick H. Mumps, measles, and rubella vaccine and the incidence of autism recorded by general practitioners: A time trend analysis. *Western Journal of Medicine*, 174(6):387–390, 2001.

[59] J. Jung. Visualizing recommendation flow on social network. *Journal of Universal Computer Science*, 11(11):1780–1791, 2005.

[60] Sugiyama K. and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995.

[61] C. Kaiser, S. Schlick, and F. Bodendorf. Warning system for online market research - identifying critical situations in online opinion formation. *Knowledge Based System*, 24:824–836, August 2011.

[62] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 107–116. ACM, 2001.

[63] F. Karimipour, M. Delavar, and M. Kinaie. Water quality management using gis data mining. *Environmental Informatics Archives*, 2:946–954, 2005.

[64] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demostration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.

[65] A. Khan, B. Baharudin, and K. Khan. Mining customer data for decision making using new hybrid classification algorithm. *Journal of Theoretical and Applied Information Technology*, 27(1):54–61, 2011.

[66] M. Khan, F. Coenen, D. Reid, H. Tawfik, R. Patel, and A. Lawson. A sliding windows based dual support framework for discovering emerging trends from temporal data. *Journal of Knowledge Based System*, 23(4):316–322, 2010.

[67] W. Kifle. Application of kdd on crime data to support the advocacy and awareness raising program of forum on street children in ethiopia. Master's thesis, Addis Ababa University, 2003.

[68] C. Kiss and M. Bichler. Leveraging network effects for predictive modelling in customer relationship management. 15th Annual Workshop on Information Technolgies & Systems, 2005.

[69] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[70] D. Knoke and S. Yang. *Social Network Analysis. 2nd Edition. Quantitative applications in the Social Sciences Series.* SAGE Publications, 2008.

[71] L. Kobus, F. Enembreck, E. Scalabrin, J. da Silva Dias, and S. da Silva. Automatic knowledge discovery and case management: an effective way to use databases to enhance health care management. In *Artificial Intelligence Applications and Innovations*, volume 296, pages 241–247, 2009.

[72] R. Kohavi. Data mining with mineset: What worked, what did not, and what might. In *Proceedings of the KDD-98 Workshop on the Commercial Success of Data Mining*, 1998.

[73] T. Kohonen. The self organizing maps. In *Springer Series in Information Science*, volume Vol. 30, 1995.

[74] T. Kohonen. The self organizing maps. *Neurocomputing Elsevier Science*, 21:1–6, 1998.

[75] K. Koperski, J. Adhikary, and J. Han. Spatial data mining: Progress and challenges. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1996.

[76] G. Krause, C. Blackmore, S. Wiersma, C. Lesneski, C. Woods, N. Rosenstein, and R. Hopkins. Marijuana use and social networks in a community outbreak of meningococcal disease. *South Medical Journal*, 94(5):482–485, 2001.

[77] V. Lampos and N. Cristianini. Tracking the flu pandemic by monitoring the social web. In *2nd IAPR Workshop on Cognitive Information Processing*, pages 411–416. IEEE Press, June 2010.

[78] H. Lauw, E. Lim, H. Pang, and Tan T. Social network discovery by mining spatio-temporal events. *Computational and Mathematical Organization Theory*, 11(2):97–118, 2005.

[79] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 915–924, New York, NY, USA, 2008. ACM.

[80] Z. Li, P. He, and M. Lei. A high efficient aprioritid algorithm for mining association rule. *Machine Learning and Cybernetics*, 3:1812–1815, 2005.

[81] P. Lingras, M. Hogo, and M. Snorek. Temporal cluster migration matrices for web usage mining. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '04, pages 441–444, Washington, DC, USA, 2004. IEEE Computer Society.

[82] A. Lipsman. The network effect: Facebook, linkedln, twitter & tumblr reach new heights in may. ComScore Inc., Jun 2011.

[83] L. Liu, S. Bhattacharyya, S. Sclove, R. Chen, and W. Lattyak. Data mining on time series: an illustration using fast-food restaurant franchise data. *Computational Statistics and Data Analysis*, 37(4):455–476, 2001.

[84] J. Malone, K. Mcgarry, and C. Bowerman. Performing trend analysis on spatio-temporal proteomics data using differential ratio data mining. In *Proceedings of the 6th EPSRC Conference on Postgraduate Research in Electronics, Photonics, Communications and Software*, pages 103–105, 2004.

[85] H. Mannila, H. Toivonen, and A. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, January 1997.

[86] P. McBurney and Y. Ohsawa. *Chance Discovery.* Advanced Information Processing. Springer, 2003.

[87] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, October 2007.

[88] J. Mennis and J. Liu. Mining association rules in spatio-temporal data: An analysis of urban socioeconomic and land cover change. *Transactions in Geographical Iinformation System*, 9 (1):5–17, 2005.

[89] P. Mika. Bootstrapping the foaf-web: An experiment in social network mining. In *1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, pages 1–10, 2004.

[90] J. Neville and F. Provost. Prediction modelling in social networks. ICWSM 2009 Tutorial, 2009.

[91] M. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:1–5, Jun 2004.

[92] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:1–15, 2005.

[93] R. Ng, J. Sander, and M. Sleumer. Hierarchical cluster analysis of sage data for cancer profiling. Workshop on Data Mining in Bioinformatics, 2001.

[94] T. Nishikido, Sunayama W., and Y. Nishihara. Valuable change detection in keyword map animation. In *Proceedings 22nd Canadian Conference on Artificial Intelligence*, pages 233–236. Springer-Verlag, 2009.

[95] P. Nohuddin, R. Christley, F. Coenen, Y. Patel, C. Setzkorn, and S. Williams. Finding “interesting” trends in social networks using frequent pattern mining and self organizing maps. *Journal of Knowledge Based Systems: Special Issue*, 29:104–113, 2011.

[96] P. Nohuddin, F. Coenen, R. Christley, and C. Setzkorn. Detecting temporal pattern and cluster changes in social networks: A study focusing uk cattle movement database. In *Proceedings 6th International Conference on Intelligent Information Processing (IIP’10)*, volume 340, pages 163–172, 2010.

[97] P. Nohuddin, F. Coenen, R. Christley, and C. Setzkorn. Trend mining in social networks: A study using a large cattle movement database. In *Proceedings 10th Industrial Conference on Data Mining, Springer LNAI*, pages 464–475, 2010.

[98] P. Nohuddin, F. Coenen, R. Christley, C. Setzkorn, Y. Patel, and S. Williams. Frequent pattern trend analysis in social networks. In *Proceedings of the 6th International Conference on Advanced Data Mining and Applications: Part I*, pages 358–369, 2010.

[99] E. Omiecinski. Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):57–69, January 2003.

[100] K. Oseman, S. Mohd Shukor, N. Abu Haris, and F. Abu Bakar. Data mining in churn analysis model for telecommunication industry. *Journal of Statistical Modeling and Analysis*, 1:19–27, 2010.

[101] S. Patton. Social networking sites: Data mining and investigative techniques. CISSP Presentation, 2007.

[102] J. Rauch and M. Simunek. An alternative approach to mining association rules. In *Foundations of Data Mining and knowledge Discovery*, pages 211–231. 2005.

[103] E. Riccomagno and J. Smith. The causal manipulation and bayesian estimation of chain event graphs. University of Warwick Publications, 2005.

[104] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70. ACM, 2002.

[105] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14:1325–1332, November 2008.

[106] J. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions Knowledge and Data Engineering*, 14(4):750–767, 2002.

[107] N. Rossol, I. Cheng, I. Jamal, J. Berezowski, and A. Basu. A real-time 3d visualization framework for multimedia data management, simulation, and prediction: Case study in geospatial-temporal biomedical disease surveillance networks. *International Journal of Multimedia Data Engineering and Management*, 2(2):1–18, 2011.

[108] C. Sabel, S. Kingham, A. Nicholson, and P. Bartie. Road traffic accident simulation modelling- a kernel estimation approach. In *The 17th Annual Colloquium of the Spatial Information Research Centre*, pages 67–75. University of Otago, Dunedin, New Zealand, 2005.

[109] J. Seifert. Data mining: An overview. Technical report, CRS Report for Congress, 2004.

[110] G. Shafer. *The Art of Causal Conjecture.* MIT Press, Cambridge, MA, USA, 1996.

[111] S. Shekhar and S. Chawla. *Spatial databases - a tour.* Prentice Hall, 2003.

[112] V. Somaraki, D. Broadbent, F. Coenen, and S. Harding. Finding temporal patterns in noisy longitudinal data: A study in diabetic retinopathy. In *Proceedings 10th Industrial Conference on Data Mining*, pages 418–431, 2010.

[113] R. Srikant and R. Agrawal. Mining sequential patterns: generalizations and performance improvements. In *Proceeding of the 5th International Conference on Extending Database Technology*, pages 3–17, 1996.

[114] O. Streibel. Trend mining with semantic-based learning. European Semantic Web Conference, 2008.

[115] V. Subramanyam Rallabandi and S. Sett. Knowledge-based image retrieval system. *Journal of Knowledge Based System*, 21:89–100, March 2008.

[116] B. Taskar, M. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems*, 2003.

[117] W. Taylor. Change-point analysis: A powerful new tool for detecting changes, 2000.

[118] K. Thearling, B. Becker, D. DeCoste, W. Mawby, M. Pilote, and D. Sommerfield. Information visualization in data mining and knowledge discovery. chapter Visualizing data mining models, pages 205–222. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[119] L. Tjung, O. Kwon, K. Tseng, and J. Bradley-Geist. Forecasting financial stocks using data mining. *Global Economy and Finance Journal*, 3(2):13–26, 2010.

[120] H. Toivonen. Sampling large databases for association rules. In *Proceedings of the 22th International Conference on Very Large Data Bases*, pages 134–145. Morgan Kaufmann, 1996.

[121] F. Tseng, C. Hsu, and H. Chen. Mining frequent closed itemsets with the frequent pattern list. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 653 –654, 2001.

[122] V. S. Tseng and K. W. Lin. Mining sequential mobile access patterns efficiently in mobile web systems. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 762–767, 2005.

[123] A. Udechukwu, K. Barker, and R. Alhajj. An efficient framework for iterative time-series trend mining. In *Proceedings of the 6th International Conference on Enterprise Information Systems*, pages 130–137, 2004.

[124] M. Vijayakumar and R. Parvathi. Concept mining of high volume data streams in network traffic using hierarchical clustering. *European Journal of Scientific Research*, 39(2):234–242, 2010.

[125] J. Wang, J. Delabie, C. Aasheim, E. Smeland, and O. Myklebost. Clustering of the som easily reveals distinct gene expression patterns: Results of a reanalysis of lymphoma study. *BMC Bioinformatics*, 3(36):–, 2002.

[126] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 2006.

[127] J. Wernecke. *The KML Handbook: Geographic Visualization for the Web*. Addison-Wesley Professional, 1 edition, 2008.

[128] M. Widmann and C. Schar. A principal component and long term trend analysis of daily precipitation in switzerland. *International Journal of Climatology*, 17(12):1333–1356, 1997.

[129] G. Williams and Z. Huang. Modelling the kdd process: The four stage process and four element model. Technical report, CSIRO, 1996.

[130] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[131] T. Wittman. Time-series clustering and association analysis of financial data, 2002.

[132] J. Wong and P. Chung. Managing valuable taiwanese airline passengers using knowledge discovery in database techniques. *Journal of Air Transport Management*, 13:362–370, 2007.

[133] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 981–990, New York, NY, USA, 2010. ACM.

[134] S. Yan, S. Abidi, and P. Artes. Analyzing sub-classifications of glaucoma via som based clustering of optic nerve images abstract. *Studies in health technology and informatics*, 116:483–488, 2005.

[135] X. Yao. Research issues in spatio-temporal data mining. In *UCGIS workshop on Geospatial Visualization and Knowledge Discovery*, pages 18–20, 2003.

[136] L. Yu, F. Chung, S. Chan, and S. Yuen. Using emerging pattern based projected clustering and gene expression data for cancer detection. In *Proceedings of the second conference on Asia-Pacific bioinformatics*, pages 75–84, 2004.

[137] W. Yuan, D. Guan, Y. Lee, S. Lee, and S. J. Hur. Improved trust-aware recommender system using small-worldness of trust networks. *Journal of Knowledge Based System*, 23:232–238, April 2010.

[138] J. Zahradnik and M. Skrbek. Classification of spatio-temporal data. In *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*, volume 2, pages 1168–1173, 2010.

[139] M. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1):31–60, 2001.

[140] S. Zemke. *Data Mining for Prediction. Financial Series Case.* PhD thesis, The Royal Institute of Technology, Sweden, 2003.