TEMPORAL OUTLIER DETECTION USING DYNAMIC BAYESIAN NETWORKS AND PROBABILISTIC ASSOCIATION RULES

By

Walter Quispe Vargas

A dissertation submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTING AND INFORMATION SCIENCES AND ENGINEERING

University of Puerto Rico
Mayagüez Campus

2019

Approved by:

Wolfgang Rolke, Ph.D. Member, Graduate Committee	Date
Marko Schutz Schmuck, Ph.D. Member, Graduate Committee	Date
Roxana Aparicio Carrasco, Ph.D. Member, Graduate Committee	Date
Sonia M. Bartolomei Suárez, Ph.D. Graduate Studies Representative	Date
Edgar Acuña Fernández, Ph.D. President, Graduate Committee	Date
Manuel Rodriguez Martinez, Ph.D. CISE Ph.D. Program Coordinator	Date

Abstract of Thesis Dissertation Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of DOCTOR OF PHILOSOPHY

TEMPORAL OUTLIER DETECTION USING DYNAMIC BAYESIAN NETWORKS AND PROBABILISTIC ASSOCIATION RULES

By

Walter Quispe Vargas

2019

Chair: Edgar Acuña Fernández
Major Department: Computing and Information Science and Engineering

Temporal datasets provide records of the evolution and dependencies of random variables over time. Recently, there has been an increase in the application of temporal datasets in areas such as intrusion detection, fraud detection, activity recognition, etc. Interesting temporal outliers are anomalies that incorporate important or new information and contradict the causal probabilistic relationship in the domain knowledge described in a temporal dataset. One main objective of Data Mining is to discover interesting temporal anomalous patterns. Moreover, provide contextualization of the interestingness of the reported outliers. Most of the methods used to discover temporal outliers are reduction-based, losing valuable information in the discovery process. On the other hand, there are scarce studies about the interestingness of reported temporal outliers. Even less, to provide contextualization of the anomaly causes.

This thesis deals with the problem of discovering these interesting temporal outliers in datasets. We present probabilistic association rules as measures to discover interesting temporal outliers based on domain knowledge that has been learned

and represented by a Dynamic Bayesian Network. Dynamic Bayesian networks are models to represent complex stochastic processes, to establish probabilistic dependencies in the feature space over time, and to capture the background knowledge in a causal relationship between features. The two probabilistic association rules:
i) *low support & high confidence*, and ii) *high support & low confidence*, were used to identify scenarios where the discrepancies between prior and conditional probabilities are significant. Our novel approach coalesces both methods. It allows us to discover interesting temporal outliers and provide contextualization in the form of relational subspaces, under the proposed methodology called "Domain Specific Temporal Anomalous Patterns."

The evaluation of the proposed methodology was done on synthetic and real temporal datasets on the unsupervised and supervised scenario. The experimental results on temporal datasets show that our approach can detect genuine temporal outliers and provide relational subspaces to explain the probable causes of the reported outliers, with reasonable efficiency measures. In this way, our technique becomes a state of the art method to discover interesting temporal outliers in temporal datasets. Designed to provide contextual information of the reported outliers; this, in turn, can be used to improve our understanding of the domain knowledge and the underlying temporal data generating process.

Resumen de la Disertación Doctoral Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de DOCTOR EN FILOSOFIA

DETECCIÓN DE VALORES ATÍPICOS TEMPORALES USANDO REDES BAYESIANAS DINÁMICAS Y REGLAS DE ASOCIACIÓN PROBABILISTICAS

Por

Walter Quispe Vargas

2019

Consejero: Edgar Acuña Fernández
Departamento: Ciencias e Ingeniería de la Información y la Computación

Los datos temporales proporcionan registros de la evolución y las dependencias de variables aleatorias a lo largo del tiempo. Recientemente, ha habido un incremento en la aplicación de los datos temporales en disciplinas como la detección de intrusos, la detección de fraudes, el reconocimiento de actividades, etc. Los valores atípicos temporales interesantes son anomalías que incorporan información importante o nueva, y contradicen la relación causal probabilística en el conocimiento de una disciplina descrito en un conjunto de datos temporales. Uno de los principales objetivos en la Minería de Datos es descubrir patrones anómalos temporales interesantes; además, proveer una contextualización de lo interesante del valor atípico reportado. Muchos de los métodos para descubrir valores atípicos temporales están basados en la reducción de dimensionalidad, perdiendo así información importante en el proceso de descubrimiento. Por otro lado, hay muy pocos estudios acerca de lo interesante de un valor atípico temporal reportado, mucho menos que proporcionen contextualización de la causa de la anomalía.

Esta tesis trata el problema de descubrir valores atípicos temporales interesantes en

un conjunto de datos. Presentamos reglas de asociación probabilísticas como medidas para descubrir valores atípicos temporales interesantes basados en el conocimiento del dominio que ha sido aprendido y representado por una Red Bayesiana Dinámica. Las redes Bayesianas dinámicas son modelos para representar procesos estocásticos complejos, para establecer dependencias probabilísticas en el espacio de variables a lo largo del tiempo y para capturar el conocimiento previo en una relación causal entre variables aleatorias. Las dos reglas de asociación probabilística definidas como: i) *soporte bajo* & *confianza alta* y ii) *soporte alto* & *confianza baja*, fueron usadas para identificar escenarios donde las discrepancias entre las probabilidades previas y condicionales son significativas. Nuestro enfoque novedoso une ambos métodos y nos permite descubrir valores atípicos temporales interesantes y proporcionan una contextualización en forma de sub-espacios relacionales, bajo la metodología propuesta llamada "Patrones Atípicos Temporales en un Dominio Específico."

La evaluación de la metodología propuesta fue realiza en datos temporales simulados y reales, en escenarios no supervisados y supervisados. Los resultados experimentales en datos temporales muestran que nuestro enfoque puede detectar valores atípicos temporales genuinos y proporcionar sub-espacios relacionales para explicar las causas probables de los valores atípicos temporales reportados, con buenas medidas de eficiencia. De esta manera, nuestra técnica se convierte en un método de vanguardia para descubrir valores atípicos temporales interesantes en conjuntos de datos temporales y diseñado para proporcionar información contextual de valores atípicos reportados, esto a su vez, puede usarse para mejorar nuestra comprensión del conocimiento de la disciplina y el proceso subyacente que genera de datos temporales.

To my parents

for encouraging me to finish my studies

for so long

so far away from home

&

To my wife

for supporting me

and giving me new dreams to pursue

&

To my son Nathaniel

for giving me a meaning in life

I love you

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Edgar Acuña Fernandez, for supporting me in my Ph.D. study and research, for his guidance, motivation, patience, and huge knowledge; for giving me so much freedom to explore and discover new areas of statistics and data mining. I appreciate all his contributions of time, ideas, and support that made my Ph.D.

I would like to thank to my committee members, Dr. Wolfgang Rolke, Dr. Marko Schütz, Dr. Roxana Aparicio and Dr. Sonia Bartolomei for your support, valuable feedback and ideas during the preparation of this thesis.

I would like to thank to my professors of CISE and Mathematics departments at UPR-RUM, Nestor Rodriguez, Wilson Rivera, Jaime Seguel, Vidya Manian and Remi Megret, for their classes and meetings have proved to be one of my best learning experiences at Mayaguez.

I would like to thank to faculty and staff of CISE and Mathematics departments at UPR-RUM, Dr. Olgamary Rivera, Sarah, Alida, Madeline, Carmen, Tania, and Zoraida, for their friendliness and support.

I would like to thank my friends and mates with whom I have had the pleasure of study and work over the years. These include Frida, Moises, Roberto, John, Velcy, Carlos, Jaime, and Hiva.

I would like to thank my wife Ysela for listen my thesis ideas, and for giving me the motivation to finish this thesis.

I would like to thank the financial support to the grant NSF:16-512 BIGDATA:CR:IA: *Large Scale Multi-parameter analysis of Honeybee Behavior*, through my advisor, that provided the funding and resources for the development of this thesis.

Contents

List of Tables

List of Figures

List of Abbreviations

BN	Bayesian Network.
conf	Confidence.
CPD	Conditional Probability Distribution.
CPT	Conditional Probability Table.
DBN	Dynamic Bayesian Network.
DMMHC	Dynamic Max Min Hill Climbing.
DSTAP	Domain Specific Temporal Anomalous Pattern.
EM	Expectation Maximization.
JPD	Joint Probability Distribution.
MAP	Maximum a Posteriori.
MLE	Maximum Likelihood Estimation.
PAR	Probabilistic Association Rule.
RS	Relational Subspace.
supp	Support.
2TBN	2-Time Slice Bayesian Network.

List of Symbols

$\mathbf{B} = (\mathbf{G}, \mathbf{\Theta})$	Bayesian network model
$\mathbf{G} = (V,E)$	Directed acyclic graph
$\mathbf{\Theta}$	Set of parameters from the BN.
X_i	Random variable
$Pa(X_i)$	Parents of X_i
$X_i[t\|$	Stochastic process
$Pa(X_i[t\|)$	Parents of $X_i[t]$
$x_i[t\|$	Temporal sequence or time series
$(B_0, B_{\rightarrow})$	Dynamic Bayesian network model
B_0	Initial network
$B_{\rightarrow}$	Transition network
R_1	Rule one: *low support* & *high confidence*
R_2	Rule two: *high support* & *low confidence*

Chapter 1 INTRODUCTION

1.1 Specifying Outliers

Outliers are data items that are substantially dissimilar from rest. Hawkins in [3] provided a highly accepted definition as: *"An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism."* Outliers are also referred to as anomalies.

Outlier detection is an outstanding research line that has been examined in different sciences areas, such as in finance to detect fraud on a credit card, in biology to monitor a disease outbreak, and in cybersecurity to avoid intrusions over a network or internet system, etc., [4]. Outliers frequently correspond to patterns in a dataset that do not behave as expected for a given domain [1, 5]. Frequently, outlier detection is a step in data pre-processing, which is essential to extract valid conclusions from data, since dirty data can lead to erroneous conclusions. This scenario is usually called "Garbage in - Garbage out." When outliers are identified by any statistical method or algorithm, they usually are removed from the database regardless of the useful information that these enclose. Currently, outlier detection is a broad field, designed to discover anomalies, and explain the interestingness of the reported outliers.

The concept of interestingness in Data Mining is related to select and rank patterns corresponding to the concern of the researcher over a specific context. Interesting or meaningful outliers are those which make sense within a specific context [6]. As an illustration of the concept of interestingness, consider the hypothetical situation described in Figure 1–1. This figure represents the dependency relation between the

income and expenditure of a group of people. The figure shows clusters of people who are similar or share the same interests. The groups C_2, C_3, C_4, C_5, and C_6 are reported outliers based on distances, dissimilarities, and densities; but contextualizing the information of data, it is evident that there exists a substantial degree of dependency on income and expenditure (generally, expenditure depends directly on the income of a person). Thus, interesting or meaningful "true outliers" are those who belong to clusters C_5 and C_6 within a specific domain knowledge, because these groups break the natural structure of dependence between the income and expenditure, e.g., is "absurd that a group of people expends more than its incomes, in a real-life scenario."

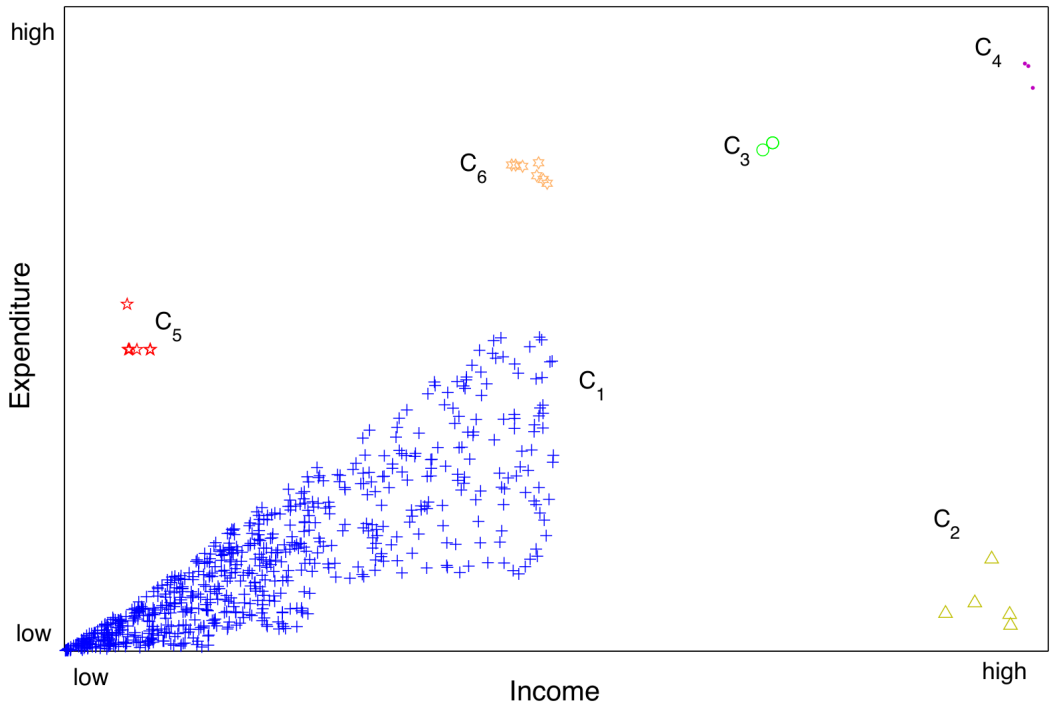

Figure 1–1: Hypothetical scatterplot corresponding to bivariate data relating to the income and expenditure of a group of people.

Analyzing the hypothetical example previously discussed, a challenging problem arises: To overcome the discrepancy among anomalies as isolated points "which are far away from their neighbors" and interesting anomalies "which make sense in a

context." This problem is subjective due to the particular interest of the researcher, and it needs an adequate method or algorithm to tackle the problem. A particular solution to this problem was presented by Babbar in [6]. The author used a Bayesian Network model in a static "non-temporal" scenario in order to understand and contextualize the information of a dataset, then, using two probabilistic association rules to uncover interesting outliers.

Non-temporal outlier detection methods, including Bayesian Networks, deal with multidimensional datasets, in which, records are treated independently of one another. These techniques are inadequate for temporal data, due to their natural dependency over the past of each variable and between variables. In this thesis, we address the temporal aspect of discovering interesting outliers, based strongly on the seminal work of Babbar.

1.2 Specifying Temporal Outliers

Temporal outlier detection methods are different from non-temporal because these do not neglect the dependency between and within variables over time. Temporal data has a degree of autocorrelation within each variable, and crosscorrelation between variables in different time lags. Thus, the methods for treating temporal data have to account for the dependency structure. In temporal outlier detection, time is an important issue, in the way to discover anomalous patterns, since it provides a natural contextualization of the problem. For example, the monthly sales of a specific toy, depends intrinsically on the month of the year (time), "usually in December, the sales of a toy are high."

The temporal anomalous patterns can occur on datasets in different ways; we can describe them as:

- Exceptional changes in a timestamp. Figure 1–2 shows a seasonal time series referred to monthly temperature measures. A temperature of $35°F$ is usually

typical in winter (at time t_1), but unacceptable in summer (at time t_2), then this will be a temporal outlier candidate. The same value of the variable temperature can be considered an outlier in the different timestamp, confirming that time is a natural contextual variable.

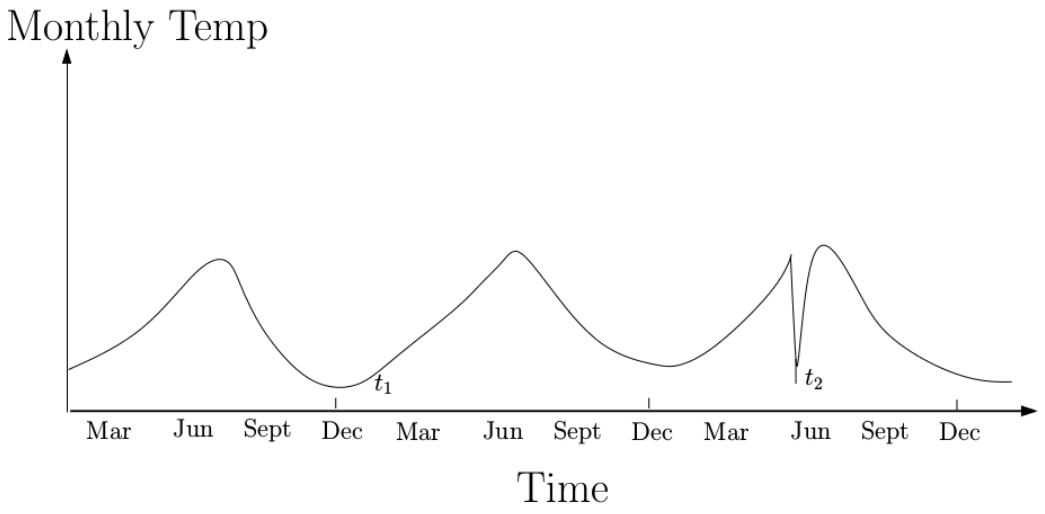

Figure 1–2: Exceptional change in a seasonal time series temperature.

- Unexpected subsequences in a time interval. Figure 1–3 illustrates a data from an electrocardiogram. The highlighted subsequence denotes a rare part of the electrocardiogram series, the low values of this part represents a rare subsequence along the time, technically it describes an "Atrial Premature Contraction." An important observation arises when each minimum value by itself does not represent a temporal outlier, but together as a subsequence will represent a temporal anomalous pattern.

- Unusual structural changes over time. For example, changes over the mean or variance of the stochastic process, usually known as a non-stationary process, occurs when trends, seasons or cycles on the data exist.

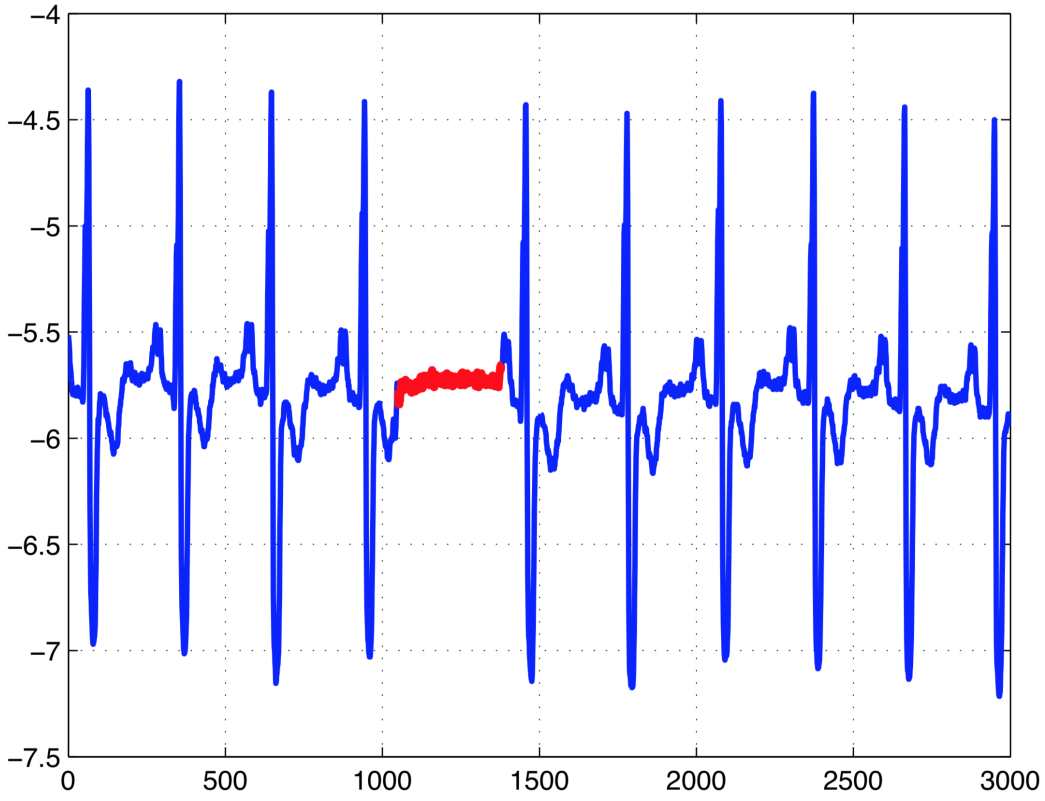

Figure 1–3: Unexpected sequences corresponding to an anomaly contraction in electrocardiogram.

These three temporal anomalous patterns described above are principal sources to uncover any temporal outlier [7]. In specific applications, such as flight safety, intrusion detection, fraud detection, healthcare, financial market, environmental science, etc., datasets are obtained with temporal structures, i.e, Discrete sequences, Discrete series, or Time Series. Therefore, the necessity to explore and develop new efficient methods to detect interesting temporal anomalies arises.

Outlier detection for temporal data is an ongoing research area. It presents different formulations due to the nature of the data, and as far as we know, it is not possible to unify the formulations for generalizing a method to detect temporal anomalies. The nature of temporal data is broad. There are different temporal sequences according to the nature of the processes, e.g., discrete sequences where the value of data is an integer number, or a time series where data is a real number. Another

classification arises on the dimensionality of temporal series, i.e., univariate or multivariate datasets. Outliers on temporal sequences are described in different ways due to multiple formulations. For example, a point into a sequence can be an outlier, a subsequence inside a sequence can be an outlier, or even the complete sequence can be an outlier corresponding to a set of normal sequences. Thus, the different formulations about temporal outliers are dissimilar from each other and are treated in different ways [8]. Since there are a limited number of methods for temporal outlier detection, the existing research has largely focused on designing measures based on reduction, distance or similarity to identify temporal outliers. However, not much effort has been invested in the concept of interestingness of outliers and the contextualization of outliers.

The occurrence of outliers in dataset is usually unknown. Sometimes an outlier is discovered as a flawed value, associated with a poor quality condition of data, then no relevant information can be captured. However, it is possible that discovered outliers will correspond to a correct instance. In this scenario, a detailed study of the existence of the outlier will represent a piece of new relevant information. In that sense, uncovering outliers is important for two reasons, first to enhance the quality of the dataset, and second to produce additional knowledge about the dataset. After uncovering outliers, instead of removing it from the dataset, appropriate analysis is necessary to explain and contextualize why such outliers emerge in the dataset and what are its possible sources.

Describing the quality of the detected outliers is a challenging task due to their subjective nature. In fact, to decide if an outlier is just white noise or incorporates new useful information is difficult. Into the last scenario, one way to decide if the outliers keep useful information is by incorporating the domain-specific knowledge [4]. Thus, the approach to establish the interestingness about the uncovered outliers will be through the well-established knowledge of a specific domain.

This thesis presents probabilistic interestingness measures to detect interesting temporal anomalies with foundations on domain specific knowledge described through a dynamic Bayesian network. The explanation of the uncovered outliers is provided. The proposed research extends the approach of Babbar in [9], to a dynamic scenario.

1.3 Dynamic Bayesian Networks

The dynamic Bayesian network model, DBN for short, represents a specific class of a probabilistic graphical model that relates probabilistic dependencies between random variables over time. Technically, a DBN model represents a set of stochastic processes, their properties, and relationships with the objectives to perform smoothing, filtering, or forecasting. The purpose of applying a DBN model to a dataset is to determine the dynamical model structure, estimate its parameters with an adequate method of estimation, and perform efficient inference. Dynamic Bayesian network models are the expansion of Bayesian network models for a temporal dataset. One characteristic of a dynamic Bayesian network is that it allows us to describe the domain knowledge inherent on the dataset. Another quality of a dynamic Bayesian network is to formulate probabilistic queries, commonly known as reasoning under uncertainty across time. Sometimes the dynamic Bayesian network model is called causality probabilistic network, due to the fact that it can handle interactions of subjective and objective information.

The dynamic Bayesian network model uses the graph structure representation to model interactions between and within variables over time, where the random variables represent the nodes of the graph, and the probabilistic causal dependency is represented by directed arrows between the nodes [10]. For example, Figure 1–4 graphically depicts a dynamic Bayesian network model, a graphical representation of a DBN model structure related to a complex stochastic process expanded on three-time instants. Within each timestamp, nodes represent random variables over

a static BN model, associated with black arrows describing the static relationship among them. On the other hand, blue arrows represent the dependency within each variable called "autocorrelation," and red arrows represent the relationship between variables called "crosscorrelation." Note that the static structure is repeated on different timestamps.

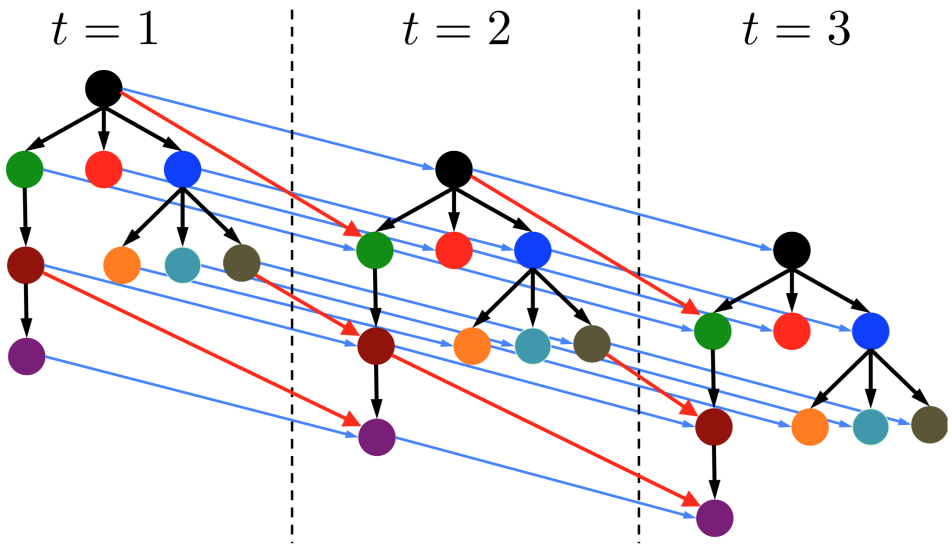

Figure 1–4: A dynamic Bayesian network that represents a complex stochastic process. Vertices depict features and arrows describe relationships.

In general, there are three crucial issues related to modeling a DBN on datasets. The first issue is to estimate the network topology e.g., estimate the tree structure in each timestamp, and the unrolled graph on different timestamps, as illustrated in Figure 1–4. The second issue refers to parameter learning, also known as the estimation process of the prior and conditional probability tables for discrete variables, or the estimation of distribution parameters for continuous variables. Finally, the inference step of a DBN model that computes probability queries of interest given some evidence, which frequently are marginal probabilities or posterior probabilities of the variables of interest [11].

The most relevant characteristic of a DBN model is its capability to encode directional probabilistic associations which can represent "cause-effect" relationships. It is conventional wisdom to consider that a good representation of a DBN model is a causal model in a domain knowledge, where causality flows in the direction of arrows on the graph [12]. The skill of a DBN model to describe the knowledge over a scheme of directed association is a significant motivation to select them as a framework; so that we can analyze observations that disrupt causal affinity, and perceptively recognize them as inherent real outliers which are reasonable within a specific context.

Based on domain knowledge captured by a BN model, Babbar in [13] indicates that "*The interesting anomalies are those data points which violate the causal semantic captured via a Bayesian network.*" Thus, a modern definition about outliers arises as "*Unlikely events under the current favored theory of the domain.*" This definition is suitably extended to a DBN model in this work. In summary, we take advantage of the work done in a static scenario to use it in a dynamic scenario.

1.4 Probabilistic Association Rules

With the aim of discovering interesting temporal anomalies, and provide explainability, it is necessary to describe the concept of probabilistic association rules (PAR), as a complementary method of the dynamic Bayesian network model.

In Data Mining, an association rule is defined as a conditional statement between random variables in a dataset. With the aim to uncover patterns, association rules are used to find the relationships between variables. In [14], a representative example is explained, "$\{milk, eggs\} \longrightarrow \{bread\}$" which represent a conditional statement, this will be declared as an association rule, the interpretation of this rule will be "*when milk and eggs are purchased, then, bread is highly likely to be purchased.*" Since an association rule will represent a frequent pattern in a dataset, measuring

the interestingness of those patterns will be necessary. The measures of interestingness have the objective to select and rank patterns in concordance with the aim of the researcher. The concept of interestingness is based on some fundamental criteria in Data Mining literature. Into the categorization of interestingness measures, the objective measures of interestingness are based on probability theory; thus, using those measures will identify objectively interesting patterns in datasets. The association rules that use objective interestingness measures are known as "probabilistic association rules." The importance of probabilistic association rules is that it can discover patterns in datasets, maintainig the uncertainty as a relevant characteristic in the discovery process. The two most used objective interestingness measures are *support* and *confidence*. The support measures the proportion that satisfies the rule; the confidence measures the reliability of the rule. Nevertheless, due to the subjective nature of interestingness, there is not an optimal measure to uncover interesting patterns [15]. Instead, we will try to reach a certain degree of interestingness and objectiveness with those measures.

Since a dynamic Bayesian network model allows a declarative relationship between parents and children under uncertainty, the probabilistic association rules can naturally be represented inside the model. In this context, two probabilistic association rules are used to find events that disrupt the declarative relationship over the DBN model. Note that the proposed methodology tries to uncover interesting temporal anomalies, which by definition, are infrequent instances. In contrast with association rules that aim to discover frequent patterns, thus the proposed rules are defined as opposite to the "causality effect" in the network. The rules are defined as:

- "*Low Support & High Confidence.*" This rule will provide patterns with low probable parent nodes, but contradictory, with high impact on the children nodes.
- "*High Support & Low Confidence.*" This rule will provide patterns with high probable parent nodes, but contradictory, with low impact on the children nodes.

The rules will uncover domain-specific temporal anomalous patterns based on the dynamic Bayesian network model learned from temporal data.

1.5 Research Contribution

In this research work, we propose a dynamic mechanism to discover interesting temporal outliers on datasets, established on dynamic Bayesian network models and probabilistic association rules for discrete sequences and time series data. The main contributions of our research are:

- Learn the dynamic Bayesian network structure and parameters from temporal datasets, with a state of the art and efficient algorithm. This step is crucial to efficiently represent the domain-specific knowledge intrinsic in the dataset, then describe the probabilistic causal dependence between random variables.
- Discover interesting temporal outliers and provide a contextualization of the reported outliers on datasets. By the methodology and algorithm called "Domain Specific Temporal Anomalous Patterns", that incorporates the learned dynamic Bayesian network and the two probabilistic association rules.

  R_1: "*low support & high confidence*."

  R_2: "*high support & low confidence*."

  This methodology provides a dynamical extension to discover interesting temporal outliers and a contextualization of the interestingness of the reported outliers, as relational subspaces composed by parents and children nodes on a specific timestamp.

The whole dynamical procedure to detect interesting temporal outliers and provide explainability is summarized as follows. First, a dynamic Bayesian network

model is learned from data, leaving us to describe both temporal structure relationships, and temporal degrees of belief. Second, the two probabilistic association rules will provide relational subspaces over a specific timestamp. In those subspaces, discrepancies over probabilistic dependencies will exist. Finally, these patterns are chosen as candidates to be interesting temporal anomalous patterns provided by their contextualization. The experimentation is performed on synthetic and real datasets. The results show that our methodology is capable of discovering interesting temporal outliers and providing contextual information.

1.6 Thesis Organization

This research work can be structured as follows: In Chapter 2, we introduce relevant works in the research of outlier detection methods, describing their qualities, advantages, and disadvantages. In Chapter 3, we present foundations about dynamic Bayesian network models, their representation, learning, and inference, as well as probabilistic association rules. In Chapter 4, we present a novel methodology that coalesces the dynamic Bayesian network model and the two probabilistic association rules to discover interesting temporal outliers, and to provide contextualization of the reported outliers on discrete sequences and time series. In Chapter 5, we present experimental results and discussion of the proposed methodology. In Chapter 6, we provide conclusions and a future research line. Finally, in Chapter 7, the ethical considerations are detailed.

Chapter 2 BACKGROUND

Outlier analysis is an important task in Data Mining and Machine Learning. Some complete surveys about anomaly detection are discussed in [1, 5, 16]. Furthermore, dedicated textbooks provide extensive treatments of outliers [3, 4]. This chapter describes the literature review related to this research work. First, we treat different aspects of the outlier detection problem. After that, we review interesing mining outliers from a static point of view. Then, we explain the current research work on temporal outlier detection. To finish this chapter, an explanation about the related research topic on this thesis, its strengths, and differences with our proposition are presented.

2.1 Aspects of Outlier Detection

An outlier is a pattern in the dataset that disagrees with the natural behavior into a specific context. Outliers are also known as anomalies; mining them is a challenging problem, due to several factors:

- The difficulty of defining the standard region where the non-anomaly data exist is a demanding task since the frontier among normal and outliers is usually not decisive.
- The lack of labeled data in order to train and validate a method to uncover outliers.
- The lack of a precise definition of the outlier through different domain knowledge.
- The challenge of reporting the quality of the uncovered outlier to determine if it is noise or if it incorporates unique, useful information.

The reasons described above make discovering outliers a difficult task. The existing techniques, methods, and algorithms solve a specific formulation within a limited context, thus there is not a unique method to consolidate the outlier detection problem. On the other hand, the formulation of the discovery process of anomalies is inferred by different characteristics such as the type of the dataset, the accessibility to the data labels, the type of detected outliers, the output of the reported outlier, and evaluation measures of outlier detection techniques. To describe them, we have the following:

Nature of Data: A data instance is characterized using a set of random variables in a dataset. The random variables could be categorical or numerical. Datasets may be univariate or multivariate; in the multivariate case, all variables might be categorical, all numerical, or in a more realistic case, a mixture of them. Another issue on the nature of data instances is the relationship between each other; for example, temporal, longitudinal or spacial data. Specifically, the temporal case, treats ordered and dependent data e.g., time series data.

Data Labels: Labels of a data instance indicate if the observation is normal or an outlier[1] . Typically, getting labeled anomalous data instances is more difficult than normal ones. Based on the condition of labels, if they are available or not, the methods can perform in the following modes:

- *Supervised Mode*: The labels are available for both normal data and outliers in a dataset.
- *Semi-Supervised Mode*: The labels are available only for normal data, but not for anomalies in a dataset.

[1] known as classes or categories of normal or anomalies instances

- *Unsupervised Mode*: The labels are not available in the dataset. In this mode, the assumption that anomalies are unusual in comparison with normal items is established.

Type of Outliers: The methods for discovering outliers are classified in agreement to Figure 2–1.

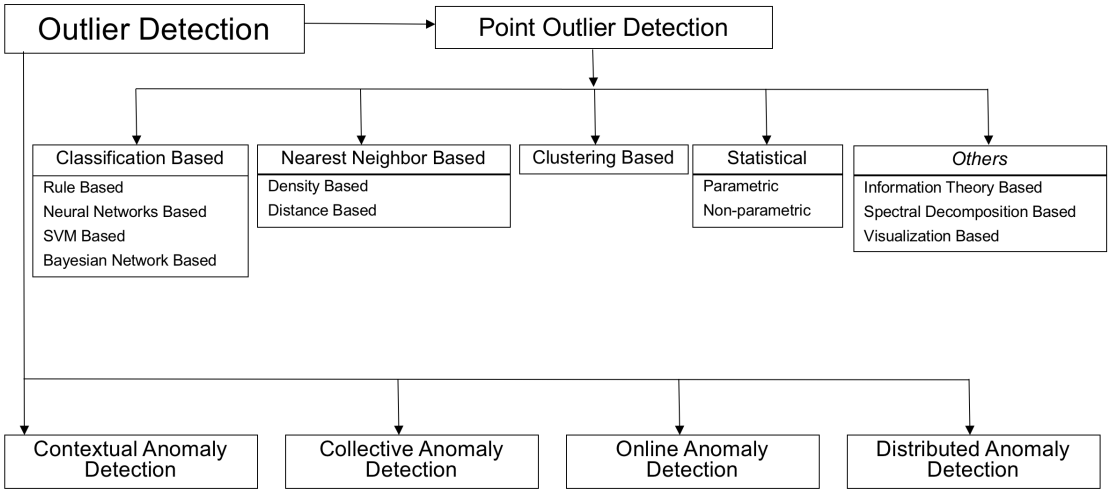

Figure 2–1: Taxonomy about the existing methods of outlier detection [1].

Most research work on outlier detection is related to point outlier detection. In this scenario, an instance is treated as unusual with respect to other data items if it is far away from clusters of normal observations. The nature of classification-based approaches relies on labeled training data, thus requiring knowledge of both normal and anomaly classes is essential. Hence, the classification of each instance or events as a usual observation or an outlier is based on the trained classifier. The main characteristics of the nearest neighbor techniques in order to discover outliers is that these techniques not require previous knowledge about the specific application domain. Clustering based approaches usually do not need data labels, so they can operate in unsupervised mode without additional knowledge about the

domain. Statistical parametric approaches require knowledge about prior distribution against non-parametric approaches. Other techniques usually require some knowledge about assumptions and measures to mine anomalies.

Contextual anomaly detection approach may uncover outliers in some specific context, but not otherwise. The context can be defined by the researcher or user in subjective mode, or the context can be learned from data. These contextual techniques usually use the concept of domain knowledge before the discovering process.

Collective anomaly detection approach analyzes data which is related to each other. The techniques need knowledge about the dependency structure of data and domain.

The rest of the techniques are relatively new in literature. There is a significant challenge to analyze online data, due to the dynamic change of common behavior, thus, the problem of updating the model to rename normal behavior of instances arises. In distributed outlier detection, since the complete process is on different branches, the high-performance computing algorithms are desired in order to perform the methods in each chunk, then merge the results on a complete integration process to report outliers.

Output of Outlier Detection: The outlier detection methods usually report scores or labels for each data instances in order to categorize an outlier as an anomaly or normal:

- *Scores*: The techniques provide a score to measure the degree of "Outlierness" to each test instance. It allows the output to be ranked but requires a threshold parameter to classify the relevant outliers.
- *Labels*: The techniques provide a data "Label" (normal or outlier) to each instance.

Evaluation of the Uncovered Outliers: The outlier detection methods are evaluated through their efficiency. This must be measured and is usually an exhaustive task since the outliers are defined as rare instances in nature. There are standard measures for evaluating them.

- *Precision*: Is the ratio among the number of properly discovered outliers and the announced outliers through the model.
- *Recall*: Also known as detection rate, is a ratio among the right discovered outliers and the complete count of outliers.
- *False alarm rate*: Defined as the ratio among total count of records in a usual class that is erroneously declared as outliers and all data instances inside of a normal class.
- *ROC Curve*: ROC stands for receiver operating characteristic. ROC curve is a compensation among recall and false alarm rate.
- *AUC-ROC*: Is the area under the curve ROC.

Different research work on point outlier detection has largely focused on developing and designing techniques to mine anomalies in databases. This research work uses dynamic version of the Bayesian network models.

The Bayesian network model belongs to classification-based techniques, thus it needs a labeled training data. However, these models can also perform in unsupervised mode in order to discover anomalies in datasets. In this case, the joint probability distribution for each record is ranked according to their likelihood, thus outliers receive a low likelihood [17].

A general approach of Bayesian networks on point outlier detection when data has different categories, is used by classical naïve Bayes network model, in order to calculate posterior probabilities of getting specific class, given evidence or test data. In this scheme, the Bayesian network model is trained in each class, then the trained

network is employed to check new data, and the class with the greatest posterior probability will be selected as the inferred class [4].

The Bayesian network model based on "*Audit Data Analysis and Mining*" (ADAM) was used to detect anomalies in the problem of detect intrusion on a network. This algorithm uses association rule methods to obtain anomalous events in the network datasets and to perform a classification in order to decide if an anomalous event is usual or an outlier [18].

In [19], a Bayesian network model was proposed to discover outliers locally in sensor data referred to the wireless scenario. It describes spatial and temporal correlations between the sensor nodes previously observed, and also describes conditional dependence between the observations of sensor variables. A Bayesian network model is trained in each node in order to detect anomalies based on the conduct of the neighboring nodes and its own measure. An instance is declared an outlier if it belongs to outside of the scope related to the predicted class.

There are many alternatives for the general approach. A relevant specific variation is to use the Bayesian network models to describe background knowledge, and use it to discover outliers in a specific domain knowledge scenario.

In [20], the researcher applied a Bayesian network model to describe the domain knowledge about a disease outbreak, and used it to detect anomalies in this context. The set of variables was separated into two user-specified classes: the variables that are associated to trends form were classified into an environmental class, and the rest of the variables constituted the indicator set. The structure of the Bayesian network model was estimated with the purpose of approximating relations only among nodes that belongs to the environmental set from the variables defined on the indicator set. The author implemented the WSARE 3.0 algorithm to contrast new data and the stated distribution described on the Bayesian network model. The objective was

to develop a guideline to uncover anomalous patterns.

Background knowledge described by a Bayesian network model was proposed in [9, 21], to take advantage of the cause-effect concept studied deeply in the scenario of Bayesian networks, as well as the concept of degree of correlation. In this context, causation was described on feature space and was captured by the existing algorithms on Bayesian scenario. The process of discovering outliers relied on applying association rules chosen from a probabilistic point of view to discover real outliers or anomalies that possessed novel information. The author defined real outliers as those which disrupted the causality effect.

In [22], the author proposes an anomaly detection method applying the Bayesian network model. The anomalies presented low likelihood events in the joint probability distribution, then the mined anomalies were evaluated to determine if they are "genuine" or "trivial" anomalies supported by user threshold.

In summary, the Bayesian network model approach is related to employing domain-specific knowledge to take advantage of uncovering true outliers. The theoretical foundations about conditional independence, and joint probability distribution enhance the method to provide a suitable explanation on why the discovered data points can be considered significant outliers.

2.2 Mining Interesting Outliers

Outliers are peculiar patterns in datasets; these patterns do not follow a normal conventional behavior in a domain. The novel theory about interestingness measures in Data Mining is oriented to ranking the discovered patterns according to the user objectives over a domain. Intuitively, interestingness is a concept of real-life importance, thus interestingness measures provide patterns which make a sense over a specific problem. In outlier detection, it is necessary to incorporate the concept of interestingness measures in order to uncover anomalies which are "real"

or "meaningful," in a specific context.

Discovering interesting outliers is a challenging problem, since there is a subjective intrinsic logic inherent on the researcher or for the subject matter specialist [23]. For example, a real life application in health surveillance is to discover a patient that has a specific disease without previous signals of symptoms, as opposed to discovering patients as an outlier that has advanced age. In the same line, when an instance is declared like an outlier, to ensure robustness and validity, a description and explanation are required to determine the specific contextualization in which it will be declared as a true outlier. Thus, this explanation will provide better understanding and interpretation of the dataset. In the health surveillance example described before, recognizing and identifying the real causes of the disease are relevant to provide adequate and timely medical treatment, inside the medical context. The existing literature in Data Mining is very limited to describe and explain outliers in a specific context [24]. The utility of outlier detection methods shows that, in many application areas, the interesting outliers are more difficult to understand and harder to find due to their nature. Many outlier detection methods described in Data Mining literature focus on detecting simple outliers that, in essence, are outliers which that not embody important information.

Interesting outliers can be uncovered and characterized by employing the domain specific knowledge. A well described domain specific knowledge of the dataset is a critical step to elaborate and develop an appropriate model in order to avoid overfitting or underfitting [4]. On the same way, the domain specific knowledge, also known as background knowledge, can be described by using the popular probabilistic graphical models, specifically the Bayesian network model or its dynamic version. As a consequence, the Bayesian network models coupled with suitable interestingness measures will establish a strong mechanism to uncover and at the same time

provide a contextual explanation about the interesting outliers.

The objective of interestingness measures is to discover patterns and rank them depending on the objectives of the researchers. Technically, the interestingness measures have been developed with the aim of evaluating the performance of association rules, therefore, specifically, interestingness measures will evaluate the performance of probabilistic versions of association rules.

The association rule is a conditional statement: $A \Rightarrow B$, A is the antecedent, B is the consequence, and both are disjoint sets of items. Suppose that D is the dataset, the association rule $A \Rightarrow B$ stands for D with the following properties, D has *support* s, and D possesses a *confidence* c. The *support* and *confidence* are interestingness measures recommended specifically for association rules. Interestingness measures for association rules, based on probability are objective in nature [14], since these achieve many of the properties to determine whether or not a pattern is declared interesting. Given an association rule: $A \Rightarrow B$, the two most important interestingness measures based on the probability are: the support described as prior probability $P(A)$, and confidence described as conditional probability $P(B|A)$.

In [6], the authors suggested using two robust probabilistic association rules which are originated in the domain causal knowledge described and learned by a Bayesian network. In order to uncover anomalous patterns for the described domain, the authors obtained patterns which fulfill either of the two rules:

$\mathbf{R_1}$:= "*low support and high confidence.*"

$\mathbf{R_2}$:= "*high support and low confidence.*"

In essence, if any instance holds with either one $\mathbf{R_1}$ or $\mathbf{R_2}$ constrained over the proposed model, the patterns can be declared as a potential candidates to be the interesting outliers using threshold parameters on support and confidence. The principal

additional issue is that they provide a problem contextualization. Such contextualization contributes a novel property into the knowledge discovery. In this line, the Bayesian network models can efficiently represent efficiently the causal-effect interaction among random variables through the concept of conditional independence with a suitable network selection. The probabilistic relationship provides a measure of uncertainty or a degree of belief between random variables or events. Thus, this quantitative measure calculates the differences in the belief which act as a sensitivity measure.

In [23], authors described the interestingness measures, and the main issues when using them as measures of sensitivity over the Bayesian network models. The authors developed an iterative algorithm in order to use many interestingness measures resulting in performance improvements and sensitivity enhancements on the Bayesian network models, thus, providing a granular method of discovering the interesting outliers.

In summary, the problem of outlier detection, and even more, mining interesting outliers in datasets, becomes a challenging task. Especially, in datasets where there probabilistic relations between random variables or causal-effects among them exist, as in the case of temporal data. Temporal datasets have qualities of data dependency; this dependency can be quantified in a probabilistic or causal manner, but it is important not to neglect this characteristic in the process of knowledge discovery. Temporal datasets have a wide range of varieties, e.g., longitudinal datasets, time series datasets, or discrete sequences datasets. In temporal datasets, the patterns will be discovered in a different manner, taking the consideration of dependency between the instances and variables as an important issue to define and uncover anomalies.

2.3 Temporal Outlier Detection

Designing data mining methods for temporal datasets represents a challenging process because of the dynamic essence of the data. This particular issue, plus the complex development of patterns, represents a difficult problem in the knowledge discovery process. The temporal aspect of mining patterns arises in different scenarios, such as sensors, medical, network, financial, etc. In such applications, the time continuity is an important issue for providing a specific order over the dataset. Thus, identifying temporal patterns over temporal continuity suggests that temporal patterns are not likely to change suddenly unless there exist anomalous factors in the dynamical processes [25]. In general, the temporal datasets have the characteristic that two consecutive values are usually close. This issue is due to the dependency structure and order of temporal datasets; thus, the pattern is not expected to change abruptly. In particular, time series datasets are a specific type of temporal data where the stochastic process is related to a continuous random variable, which we will study in this research.

A time series is a sequence of values in chronological order. Technically, a time series is a realization of a stochastic process. The set $S = \{x_i[1], x_i[2], ..., x_i[t], ..., x_i[T]\}$ represents a time series, where the index $t = 1, 2, ..., T$, represents time, and the subindex $i = 1, 2, ..., n$ describe each variable. Thus, $x_i[t]$ is a data instance i defined in the instant time t. When $n = 1$, the time series S will represent univariate time series; for $n > 1$, the time series S will be multivariate or multidimensional.

Anciently, in order to discover temporal outliers in a multivariate scenario, an equal dimension of each series was needed, i.e., an equal number of observations over time, and same time granularity (spaced time) over temporal data. Currently, data mining methods to detect outliers on sparse or irregular temporal multivariate data are challenging novel research topics.

The discrete sequences are another important class of temporal data that we treat

in this research work. These discrete sequences are characterized as a sequence of events, which are represented by tags if they belong to an alphabet composed of symbols. In formal language theory, the symbols form an alphabet; for example, the events in sequences related to the DNA molecule are symbols that form the alphabet $\{A, C, G, T\}$. In the same line, there exist other more complex alphabets, in which complex discrete sequences will compound the dynamic process.

Note that discrete sequences can occur in different timestamps, not necessarily equally spaced as in time series. However, both time series and discrete sequences can be related through a discretization process, in order to conduct a proper analysis. The discretization of a continuous real-valued time series can be transformed into qualitative (categorical) data or discrete sequence. With these important class of temporal data, we describe different methods of outlier detection.

The lack of continuity concerning recent neighbors or its past, defines a temporal outlier [4]. To describe an example, the sudden changes in temporal data will represent a temporal anomaly. Another example is if the different appearance of a subsequence into the whole temporal sequence will represent a temporal pattern outlier. Temporal outlier detection examines anomalous patterns across time. It uses the concept of temporal continuity in order to mine unusual changes (an abrupt change in the trends), unusual sequences, and unusual temporal patterns (structural changes). Different aspects of temporal data type should be considered, for example, if the detection analysis is performed on time series, discrete sequences, data stream, space-temporal data, etc. If the analysis is proposed on univariate or multivariate datasets, it is another crucial aspect. The present research is about multidimensional scenario, e.i., given a multivariate time series or discrete sequence, find point outliers or sequences (collective) outliers. On the other hand, an essential strategy to mine temporal outliers is founded in the concept of windows, in this process, the temporal series will be fragmented in chunks or windows with a set

length, the analysis strategy is to treat each window as a unit or item.

In temporal outlier detection, new challenges arise, for example, for diverse applications, it may often not be possible to use a standard model, since, there exist ample alternatives to the problem statements. This reason is essential to use the dynamic Bayesian network models due to the flexibility to represent temporal data in an equidistant order (time series) or in different timestamps (discrete sequences). Due to the complex dynamic nature of the data, modeling this scenario is a challenging task. The dynamic Bayesian network model can handle this complex dynamic data since it supports categorical or numerical data. Also, a dynamic Bayesian network model has the property to update its parameters (probabilities) in different timestamps [10]. As mentioned in the case of non-temporal, judging the quality of the reported outlier is a challenging problem. Determining if an outlier is noise or embodies new information, and contextualizing the problem is vital. In this line, the use of background knowledge is required. One way of ascertaining the quality of the reported outlier is by using dynamic Bayesian network models. The DBN probabilistically describes background knowledge and is capable of contextualizing the information in temporal datasets.

As far as we know, regarding temporal outlier detection in the scenario described above, some relevant research work on temporal outlier detection can be found in [8]. Here, the author develops an anomaly detection techniques based on semi-supervised mode related to data sequences, precisely time series data in a univariate and multivariate scenario. The main approach is related on windows methods, in the scenario when data are symbolic sequences, applying an alternative model of finite-state automaton in order to set scores. Mainly, low scores on absent states in that the outliers sequences represent more plausible to include those states, compared with normal sequences. In the scenario of anomaly detection on univariate

time series, the adoption of methods of support vector regression coupled with nearest neighbor density to discover outliers. The method is called $WINC_{SVM}$ and improves the performance of the discovery process compared to related methods. As a consequence, anomaly detection on multivariate time series is performed by employing subspace monitoring for converting a multivariate stochastic process to univariate by encapsulating the intrinsic dynamic nature of the data. Finally they applied the previous $WINC_{SVM}$ adaptation.

In [26], independent component analysis (ICA) is proposed with the aim of finding point anomalies. Those anomalies are also known as novelties in multivariate time series datasets. The approach is to project the multivariate time series to an independent component, where each component is treated as univariate one, reporting signals of the anomalies, to assess the outlier signal compared with threshold parameters, to discover outliers. The ICA model considers the linear combination of independent components and independent noise in the observed signals; a supplementary assumption is that the noise possesses a high kurtosis measure.

One statistical method to project a multidimensional time series on interesting projections is the well-defined projection pursuit. In [27], the authors adapted the projection pursuit techniques and implemented algorithms in order to discover anomalies on time series in a multi-dimensional scenario. The method selects an interesting direction projection to uncover the outliers in this right projection. The procedure for selecting interesting direction projection was based on the kurtosis coefficient, using numerical optimization to find the coefficient. Finally, to discover anomalies, the authors used the mentioned interesting direction projection.

In the scenario of discovering anomalies based on graphs, the concept of kernel matrix is exploited in [28], In this work, authors use an extension method known as kernel matrix alignment. This novel method can characterize the relations between variables that describe each time series. They developed an efficient algorithm based

on the traversal random walk inside the graph that was derived for the extension method kernel matrix alignment to uncover temporal anomalies in the multivariate scenario. The algorithm can detect point and subsequence anomalies.

Many research works on detecting temporal outliers in a multivariate scenario are reduction based techniques. In summary, these methods take a multivariate time series or discrete multivariate sequence as an input. These are transformed into a univariate one. Finally, an algorithm is used to detect anomalies in a univariate scenario. This approach, naturally, loses essential information in the discovery process. Thus new techniques will be needed in order to capture meaningful outliers and provide a contextualization of them.

2.4 Related Work to the Proposed Problem

Into the scenario of uncovering significant temporal anomalies and providing the explainability of them, background knowledge is a central issue in the process of discovery. A way of describing the background knowledge or domain-specific knowledge is through the dynamic Bayesian network model. This particular problem statement has not been solved in the current literature yet. However, some research works tried to solve the problem, but the complexity is a bottleneck problem. Some of the most related to our proposal is described as follows:

In [29], a couple of algorithms to uncover outliers in sensor data based on the dynamic Bayesian network model were presented. The first strategy used a hidden Markov model (a specific class of a DBN model). In this, the author used a Kalman filter model in order to gradually get the posterior distribution, related to hidden state and observed state, in the case when new measurements exist and are accessible. Using the posterior distribution from the observed state, the Bayesian credible

intervals for the latest measurements were generated. Thus, if some measurements do not belong to a Bayesian credible interval, then it is declared an outlier. The second strategy is to apply the well-known 2-layer dynamic Bayesian network model. The hidden state of this model is related to the label normal or an outlier in each measurement derived from the sensor. The maximum a posteriori is computed, related to the hidden states that indicate the status of the measurement, to characterize if a measurement is normal or an outlier.

In [30], a procedure based on a classification model was performed to uncover outliers on "large video sequences of the laser superficial heat treatment process of steel cylinders." Before the discovery process, they selected some relevant features based on clustering algorithms. The outlier discovery process (in situ) applies dynamic Bayesian network algorithms for two purposes. First, characterization of the temporal process. Second, using a some of the structure learning algorithms to represent a usual process. The process of uncovering anomaly sequences of consecutive frames is performed based on the anomalies scores, obtained from the log-likelihood of sequences over the dynamic Bayesian network model.

In [31], authors proposed an algorithm for a contextual type of pilot error detection applying dynamic Bayesian networks as a framework tool in order to learn the model and discover temporal outlier instances. The dynamic Bayesian network topology was described and learned based on the actions of the pilots and the data records from sensor instruments into the flight scenario. The anomalies appeared in both processes: classification and prediction. The outliers usually pose a wave effect consequence over the next items on the identical timestamp and further timestamps. Consequently, if the outlier is discovered on a specific instant, its effect will extend to the connected cases on the identical time instant and also on the next instants. Eventually, the effect will disappear in the short term and the instances will return to normal. Thus, the longer the outlier happens, the extended and higher the effect

will be.

In [11], two particular cases of the dynamic Bayesian network models to detect anomalies in flight datasets were proposed. The first DBN model, hidden semi Markov, was used to represent discrete sequences. The authors developed and presented an efficient algorithm based on a spectral scenario in order to perform the inference process in this model. The second DNB model was a vector autoregressive to represent time series on the multidimensional scenario. A similarity neighborhood graph was constructed to uncover outliers and determine the event of anomalies. Finally, the author combined semi Markov and autoregressive models for representing multidimensional mixed time series. The model proposed was the semi Markov switching vector autoregressive. In order to discover outliers in a flight scenario, a prediction-based model was used, measuring dissimilarities on prediction and observation.

The research works aforementioned, use the dynamic Bayesian network model as the central tool to describe domain knowledge. However, those previous works do not learn the structure from datasets. Instead, the dynamic structure was fixed based on subjective expertise. Thus, previous research works uncover anomalies in temporal datasets, but they do not uncover meaningful or real outliers. Instead, they provide regular outliers. Finally, another drawback of those methods is that they do not provide contextualization of the reported outliers. Thus, we propose to enhance the use of dynamic Bayesian network models by providing them probabilistic association rules to discover and contextualize interesting temporal outliers.

Chapter 3 DYNAMIC BAYESIAN NETWORK MODEL AND PROBABILISTIC ASSOCIATION RULES

This chapter describes concepts of the dynamic Bayesian network model, in order to mine interesting anomalous patterns of temporal data. Given a temporal dataset about the specific stochastic process, for example, a protein sequencing to determine the amino acid subsequence or the whole protein, a stock market to make efficient forecasting, etc. Depending on the interest of the researcher, there exists a general need to develop a model to represent the temporal data in order to describe its properties, discover patterns, mining rules, or perform forecasting. One of the most suitable models is established on statistical and probability theoretical models, which are usually created by random processes through datasets. The mentioned models describe the dependency structure among random variables in an efficient way. Moreover, these models can handle many parameters related to the research problem and can efficiently describe the evolution of the features over time. In this scenario, to efficiently describe relationships between variables, the Bayesian network model is frequently used. This model is a particular case of probabilistic graphical models to characterize dependencies among variables in a static scenario. The dynamic extension is introduced as a temporal ingredient. The extension is provided to the static Bayesian network models, thus arises the well-studied temporal model known as the dynamic Bayesian network model [12].

The organization of this chapter is as follows: First, since the dynamic Bayesian network model is defined as a dynamic version of the Bayesian network model, we present necessary concepts on the Bayesian network models. Second, essential

definitions of the dynamic Bayesian network model: representation, learning, and inference are presented. Third, we provide some useful concepts about probabilistic association rules to merge it with the dynamic Bayesian network model to uncover and contextualize interesting temporal outliers. Finally, we describe a discretization method for time series.

3.1 A Bayesian Network as Graphical Guideline

The Bayesian network model represents one particular case of the well-known probabilistic graphical models. The Bayesian network model, also defined as a belief network, combines the theory of probability and graph theory to substantially describe problems in which there exist issues like uncertainty and dependency. Into the Bayesian network model, the graph topology has the capability of encoding the domain-specific knowledge, through the directed relations between the nodes which represent random variables attached with arrows beginning in a node known as parent and finishing in other node known as a child.

In the Bayesian network model, the probabilistic relationship among random variables is usually described as a "cause-effect" framework [32]. Two components are required to establish a Bayesian network model, the qualitative or structure component, and the quantitative or parameter component. The qualitative part is related to find variables to built up the structure of the graph topology; the graph is formed by relating two random variables as long as they have a probabilistic relation. Note that the Bayesian network model will be a directed acyclic graph since it has a unidirectional relation as a cause-effect relation. The quantitative part is related to estimate parameters of the distribution of a given random variable. The parameters are probabilities in a discrete case, and statistical parameters in a continuous case; in both cases, the parameters represent the degree of relationship between variables that are connected by an arrow in the Bayesian network model.

The Bayesian Network models are related to the subjectiveness of the research problem. The Bayes theorem is used to update the parameters using previous or new information. The Bayesian network model can differentiate the causal and evidential scenarios in the learning and inference process. These described qualities from the Bayesian network model were established on foundations in statistics, probability, and information theory. Nowadays, it is widely used and applied in Data Mining, Machine Learning, Artificial Intelligence, and Data Science [32].

3.1.1 Definitions and Properties

A *Bayesian network* model is a couple of the form $(\mathbf{G}, \mathbf{\Theta})$, where each component are described as:

- $\mathbf{G} = (V,E)$ is a *Directed Acyclic Graph* (DAG), where, V represent a collection of nodes (also known as graph vertices), to represents *random variables*, thus $V = \mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. On the other hand, E represent the collection of edges (oriented arcs); each edge characterize the probabilistic dependencies among random variables, thus $E = \{(X_i, X_j) | X_i, X_j \in V\}$, where each arrow (X_i, X_j) is defined as: If X_i then X_j, or $X_i \rightarrow X_j$, or the characterization: X_i "*is parent of*" X_j, or X_j "*is child of* " X_i.
- $\mathbf{\Theta}$ represents the set of parameters from the Bayesian network model, that provides the quantification of the graph model. In the discrete case, the set of parameters $\mathbf{\Theta} = \theta = \{P(X_1|Pa(X_1)), \ldots, P(X_n|Pa(X_n))\}$ represents the set of *conditional probability distributions* (CPD), where, $P(X_i|Pa(X_i))$ is a conditional probability distribution from a child X_i given its parents $Pa(X_i)$; it is also called conditional probability table (CPT). In the continuous case, the set of parameters $\mathbf{\Theta}$ will represent the set of parameters that characterize the distribution of a random variable. If a random variable has a Gaussian distribution with parameters μ and σ in the

network model, then, the set $\boldsymbol{\Theta}$ will have the elements μ and σ.

There are fundamental concepts in Statistics and Graph theory to understand the Bayesian network model; we describe it in the following.

Conditional independence assumption or directed local *Markov assumption* is defined as for each random variable X_i, it is probabilistically independent of its Non-descendants, given its parents, e.i.

$$\forall X_i \in \mathbf{X}, X_i \perp NonDes(X_i)|Pa(X_i). \tag{3.1}$$

When the nodes of the graph are ordered topologically (Parents before Children), then a specific node is declared as a descendant of a node if the former is positioned after the latter in the graph structure.

The chain rule property in a Bayesian network model states that the joint probability distribution (JPD), will be factorized due to the assumption 3.1, e.i.

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)) \tag{3.2}$$

The factorization provides enormous simplification on the computation of the JPD since a node on a graph depends uniquely on its progenitors.

Information flows in the Bayesian network model occurs when there is new evidence from some random variables on the model, and someone is interested in compute some posterior probability referred to another subset of features given available evidence. Thus, the information in a Bayesian network can drift in three different kinds according to the way of connections of the nodes in the graph structure.

- Serial connections are often addressed as causal chains, e.g., $X_1 \rightarrow X_2 \rightarrow X_3$. Here, if there is no evidence, the information flows in both directions through the nodes in this connection. A particular situation happens when the middle node is available; then, the information flows between the extreme nodes may not be available in this scenario.
- Diverging connections have a common cause, e.g., $X_2 \leftarrow X_1 \rightarrow X_3$. Here, like in previous connection, the information flows despite there is not any evidence.
- Converging connection or *v-structure*. This connection is characterized by a single destination node of two or many nodes described as parents, e.g., $X_1 \rightarrow X_3 \leftarrow X_2$. The main difference between previous connections is that here, the information cannot flow among variables if there is not any evidence about the destination node.

d-separation in the Bayesian network model: Given X and Y nodes on a graph, If there exists another node Z on any path among X and Y (could be undirected path) over the graph, thus it says that "X and Y are *d-separated*, such that Z satisfy" either:

- Node Z is known, and it is located on a serial or diverging connection on the graph.
- Node Z is located on a converging connection on the graph, and neither Z nor any descendants of Z have got evidence.

Figure 3–1 graphically depicts these two cases. The *d-separation* concept represents the properties of probabilistic conditional independence in the topology graph. The probabilistic conditional independence is intuitively defined on the graph theory as: If the random variables (nodes) X and Y are *d-separated* by another random variable (node) Z, then X and Y are *conditionally independent* given Z [12].

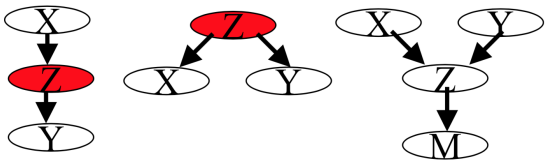

Figure 3–1: d-separation cases.

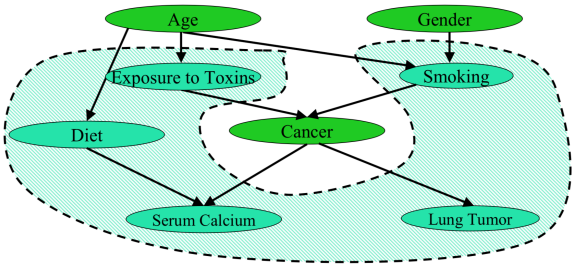

Figure 3–2: Markov blanket.

The Markov blanket in the Bayesian network model. Given a node X in the graph, the *Markov blanket* of X is a collection of parents, children, and all nodes that share a common child of X. The Markov blanket describes a subset of nodes such that completely *d-separated* a node of other nodes into the graph; thus, given a Markov blanket from a node, this node is independent of other nodes in a conditional way [12]. Figure 3–2 shows a hypothetical Bayesian network graph and a Markov blanket of the node cancer in dashed lines.

Independence map: The independency map (**I-map**) represents the agreement among *graphical separation* which is defined as the lacking of an arc between nodes, and *probabilistic independence* which describes the degree of association between random variables [32]. The combination of graph and probability theories will represent a DAG with probability distribution **P**, as an **I-map**. The probabilistic conditional independence in **P** will represent the concept of *d-separation* in the **I-map** graph, e.i., for every **X**, **Y**, and **Z** subsets of nodes of DAG. If **X** and **Y** are *d-separated* given **Z**, then **X** and **Y** are conditionally independent given **Z** in the distribution **P**. In simple words, the network topology characterizes relationships between variables in a conditional independence manner. As a consequence, an alternative definition states that a Bayesian network model is a minimal **I-map**, in which none any arrow can be deleted from the graph structure without invalidating the I-map property [12].

The described properties provide foundations to perform modeling in the Bayesian network from a dataset. Thus, the Bayesian network model represents a complete framework with the aim of learning and inference processes. On the other hand, there are different types of Bayesian network models, depending on the type of random variables. Thus, we can describe it as follow:

Types of the Bayesian network models: Due to the nature of variables (discrete, continuous, mixture), and the type of distributions that they follow (Multinomial, Gaussian, etc.). The Bayesian network model can be categorized as discrete, continuous, or hybrids. An important type of Bayesian network model is the discrete Bayesian network, where the conditional probability distributions for all variables are multinomials. Another important class is the Gaussian Bayesian network, where the conditional probability distributions of all variables are linear Gaussian [12]. The hybrid Bayesian network model where any discrete variable cannot have continuous Gaussian parents are called linear conditional Gaussian networks (LCG).

The main objectives of the Bayesian network model from datasets are to perform learning and inference process; in the following, we describe both:

Learning and Inference in a Bayesian network model: Learning is de process to fit a Bayesian network model from data, it is performed in two different steps, *structure* and *parameter* learning [33]. The inference is the process of answering probabilistic questions in a Bayesian network model when there is a piece of new evidence. The relevant research work inside the learning process on Bayesian network models are described in [34]. This work describes contemporary research works, and some references state that the process of estimate the topology of a Bayesian network model has an "NP-hard computational complexity."

3.1.2 Structure Learning

The process of discovering the topology of the graph for a Bayesian network model is called structure learning. The principal objective of estimate the graph skeleton for a Bayesian network model represents the probabilistic conditional independence between random variables in a graphical dependency through the directed arrows. By definition, the graph structure of a Bayesian network model must be a minimal I-map in order to get the dependency graph topology of the dataset, or at least the graph structure must be close to the actual probability distribution of the nodes. The procedures to identify the graph structure are categorized in a couple of categories. The first is related to discovering the graph structure of the network by computing independence tests in a restricted mode; these independence tests are performed over the nodes. The second is based on the searching process. Search an adequate graph structure over the set of all possible networks composed by the predefined nodes; this process mainly uses scores to optimize a specific criterion and also uses many search algorithms, like the heuristic greedy search. As a natural consequence of both; the hybrids methods can be considered, then we can consider three categories of algorithms for structure learning of a Bayesian network model [35]. In the following, we describe the three mentioned categories.

Constraint-Based Algorithms: The constraints are usually conditional independence statements. The regular tests to determine the conditional independence between nodes are applied in a real scenario and are theoretical tests from a statistical perspective based on evidence. The disadvantages with this approach are: First, this approach is hard to discover reliable the properties related to probabilistic conditional independence. Second, it is difficult to find the network topology structure optimally, even more, the algorithms are susceptible to tests of independence between nodes, in the presence of failures. If we got wrong answers on the

procedure of the independence test, then, the network construction procedure will be wrong [36].

The two first algorithms that we describe are SGS and PC algorithms. The former, establish if there exists an arrow among a couple of nodes, through of independence test restricted overall subsets composed by other nodes. The latter, makes tests based on independencies among all pairs of variables conditioned over other subsets composed by other variables ordered from little to big. The other algorithms (GS, IAMB, fast-IAMB, inter-IAMB) are based on the Markov blankets; first, determine the blanket of each node, simplifying the search over the blanket to determine the existence of edges [35]. As a remark; usually, this approach line has been a favorite selection of researchers with a focus on estimate "*causal models*" from datasets.

Score-Based Algorithms: These procedures consist in to provide a score to each Bayesian network structure candidate; this score measures the goodness of fit of the Bayesian network model that best describes the dataset. The main disadvantage of this class of algorithms is to compute the scores for all candidates graph structures. To alleviate this problem is necessary to design greedy algorithms to possible find suboptimal graph structures for candidates [36]. In general, the score-based algorithms usually are characterized by statistical inference concepts, like the well-known minimum description length, or the score based on Bayesian statistics. The score that frequently is computed is the BIC-score, BIC stands for *Bayesian information criterion*; this score is compound by the likelihood of the graph structure, and the penalty term to control the complexity of the network model. BIC-score will be derived from posterior probability relating to the network structure.

Consider a dataset D, assuming that the network topology G is a random variable with prior probability distribution $P(G)$, then by *Bayes theorem*, we obtain the

posterior probability as:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{3.3}$$

In order to find the BIC-score, we only need to maximize the numerator of the fraction because the denominator does not depend on G. To get the likelihood of dataset given a network topology $P(D|G)$, in Bayesian statistical scenario, the usual way is to averages summarizing on the set of all feasible parameters, ponderating each parameter through its posterior probability distribution:

$$P(D|G) = \int P(D|G, \mathbf{\Theta})P(\mathbf{\Theta}|G)d\mathbf{\Theta} \tag{3.4}$$

A difficult stage is to compute the previous integral if the prior probability does not conjugate with likelihood. To avoid full calculation of the integral in 3.4, examine its asymptotic behavior is essential. If the sample is large in the limit, it is said that "the posterior probability is robust to the selection of the prior probability" [33]. *Schwarz* showed that the asymptotic estimation for appropriate (does not give zero probability of any event) priors.

$$\log P(D|G) = \log P(D|G, \hat{\mathbf{\Theta}}) - \frac{d}{2}\log N \tag{3.5}$$

Where the estimator $\hat{\mathbf{\Theta}}$ represents the estimation of parameters on G in order to optimize the likelihood distribution from dataset, d is defined as the dimension of G (number of parameters). Finally, the penalty term is $-\frac{d}{2}\log N$, this penalty act as a trade-off measure between overfitting and underfitting; specifically, it neglects complex structures; thus, it avoids that the model will fall in overfitting.

In order to provide a ranking of all possible networks structure candidates, the BIC-score applies 3.5 without parameter priors.

Some algorithms of score-based are *hill-climbing* and *K2*. Those techniques usually seek in the space of possible structures greedily, beginning with a null network,

then adding, deleting, or reversing an arrow one at a time until reaching convergence on the score or is not possible to improve the score [35]. Finally, the main drawback of the score-based approach is that, may have to search in an extensive set of possible graph structures, making it computationally infeasible.

Hybrid or local search algorithms: This class of algorithms merges the constraint and score algorithms to balance its disadvantages. In literature, the two best-known algorithms are the *Sparse Candidate* and the *Max-Min Hill-Climbing*. Both procedures perform in two phases: restriction and maximization. The first phase restricts the possible set of candidates for the role of parents en the node X_i, which made smaller from the complete set $\mathbf{X}$ to a subset of nodes $\mathbf{C}$, in which the behavior showed the relation with X_i. In the second phase, given a score function, the maximization process optimizes these function, according to imposed constraints $\mathbf{C}$. In the *Sparse Candidate* algorithm, the maximization process will be done iteratively in two steps until reaching an optimal network score, under some parameters threshold previously specified [37]. In the *Max-Min Hill-Climbing* algorithm, the maximization process performs these two steps only once by using a subroutine called "*Max-Min Parents and Children*." This subroutine will be performed to estimate the nominee sets $\mathbf{C}$ into the heuristic model, in order to find the optimal network topology [38].

3.1.3 Parameter Learning

After the topology structure of a Bayesian network model was estimated, then, the task now is learning (estimate) the parameters of the network. By the chain rule property described in 3.1.1, learning parameters are greatly simplified, since the joint probability distribution can be factorized by local distributions, that in a real situation usually has a few numbers of parameters. In literature, there exist a

couple of perspectives to the process of learning the parameters: The well-known *Maximum Likelihood Estimation* (MLE) for short, and the Bayesian estimation process through maximum a posteriori (MAP) for short; or full Bayesian scenarios. The problem setting is:

Let $\mathbf{B} = (\mathbf{G}, \mathbf{\Theta})$ be a discrete a Bayesian network model with known structure; we have n random variables, $X_1, X_2, \ldots, X_n$; each variable, let us say X_i with r_i states; the number of configurations of $Pa(X_i)$ equal to q_i, then the parameters to be estimated are:

$$\theta_{ijk} = P(X_i = j | Pa(X_i) = k), \quad i = 1, \ldots, n;\ j = 1, \ldots, r_i;\ k = 1, \ldots, q_i \tag{3.6}$$

The parameter vector $\theta = \{\theta_{ijk} | i = 1, \ldots, n;\ j = 1, \ldots, r_i;\ k = 1, \ldots, q_i\}$.
The vector of parameters for $P(X_i|Pa(X_i))$, $\theta_{i..} = \{\theta_{ijk} |\ j = 1, \ldots, r_i;\ k = 1, \ldots, q_i\}$.
For vector of parameters $P(X_i|Pa(X_i) = k)$, $\theta_{i.k} = \{\theta_{ijk} |\ j = 1, \ldots, r_i\}$.
Note that $\sum \theta_{ijk} = 1; \forall i, k$. Let D be the complete dataset with m rows and n columns, each row, usually called a data instance; the log-likelihood is:

$$l(\theta|D) = \log L(\theta|D) = \log P(D|\theta) = \log \prod_{l=1}^{m} P(D_l|\theta) = \sum_{l=1}^{m} \log P(D_l|\theta) \tag{3.7}$$

Consider the function $I(i, j, k : D_l) = 1$ if $X_i = j, Pa(X_i) = k$ in D_l, this represents the characteristic function of instance D_l; and $m_{ijk} = \sum_l I(i, j, k : D_l)$, then, making some calculus we obtain.

$$l(\theta|D) = \sum_{l=1}^{m} \log P(D_l|\theta) = \sum_{i,k} \sum_{j} m_{ijk} \log \theta_{ijk} \tag{3.8}$$

We are trying to compute:

$$\arg\max_{\theta} l(\theta|D) = \arg\max_{\theta_{ijk}} \sum_{i,k} \sum_{j} m_{ijk} \log \theta_{ijk} \tag{3.9}$$

Optimizing equation 3.9, we can obtain the MLE for θ_{ijk}, as:

$$\theta^*_{ijk} = \frac{m_{ijk}}{\sum_j m_{ijk}} \tag{3.10}$$

In words, the MLE for $\theta_{ijk} = P(X_i = j|Pa(X_i) = k)$ is:

$$\theta^*_{ijk} = \frac{\text{number of cases where } X_i = j \text{ and } Pa(X_i) = k}{\text{number of cases where } Pa(X_i) = k}$$

In a Bayesian estimation, we view θ as a vector of random variables with prior probability distribution $P(\theta)$, then by *Bayes theorem*, $P(\theta|D) \propto P(\theta)L(\theta|D)$. From 3.8, we obtain the posterior probability.

$$P(\theta|D) \propto P(\theta) \prod_{i,k} \prod_j \theta_{ijk}^{m_{ijk}} \tag{3.11}$$

Using a conjugate Dirichlet prior, for $P(\theta_{i.k})$, e.i. $\theta_{i.k} \sim Dir(\alpha_{i0k}, \alpha_{i1k}, \ldots, \alpha_{ir_ik})$, where α_{ijk} are the hyperparameters. Thus, $P(\theta_{i.k}) \propto \prod_j \theta_{ijk}^{\alpha_{ijk}-1}$, neglecting some constants. Then, we have the product of Dirichlet distributions as prior.

$$P(\theta) = \prod_{i,k} \prod_j \theta_{ijk}^{\alpha_{ijk}-1} \tag{3.12}$$

Replacing 3.12 in 3.11, the posterior distribution is a product of Dirichlet distributions.

$$P(\theta|D) \propto \prod_{i,k} \prod_j \theta_{ijk}^{m_{ijk}+\alpha_{ijk}-1} \tag{3.13}$$

The posterior predictive for a new data instance D_{m+1} is:

$$P(D_{m+1}|D) = \prod_i P(X_i|Pa(X_i), D) \tag{3.14}$$

Further, we have

$$\begin{aligned} P(X_i = j|Pa(X_i) = k, D) &= \int P(X_i = j|Pa(X_i) = k, \theta_{ijk}) P(\theta_{ijk}|D) d\theta_{ijk} \\ &= \int \theta_{ijk} P(\theta_{ijk}|D) d\theta_{ijk} \end{aligned} \tag{3.15}$$

From 3.13, we have $P(\theta_{i.k}|D) \propto \prod_j \theta_{ijk}^{m_{ijk}+\alpha_{ijk}-1}$, then 3.15 turns in Bayesian estimation closed-form:.

$$P(X_i = j|Pa(X_i) = k, D) = \frac{m_{ijk} + \alpha_{ijk}}{\sum_j (m_{ijk} + \alpha_{ijk})} \tag{3.16}$$

Thus, 3.16 is the Bayesian maximum a posteriori (MAP) estimate of θ, such that $\theta^* = \arg\max_\theta P(\theta|D)$.

Estimating parameters when D is uncompleted, makes MLE no longer a viable option. Deal with missing data, specifically the missing at random (MAR); an assumption is required; the assumption states that, if in the data are features with MAR, the lost values of the features depend on the rest of features which are observed totally on data. In this line, the standard approach for estimating parameters is the well-known *Expectation-maximization* (EM) algorithm; this procedure supposes that the missing values will be in the MAR scenario. The EM procedure begins with an elementary estimation θ^0, then, in each loop t, in the expectation step fills gaps in the dataset based on θ^t. In the maximization step, after complete gaps, the estimation process is required in order to recompute parameters to obtain θ^{t+1}. The EM procedure continues in a loop mode until to reach convergence to a maximum. The EM algorithm is usually fast, especially at the first few iterations. Moreover,if there exists a larger amount of missing data, the convergence will reach slower. Finally, there is no guarantee that the EM algorithm converges to the global optimum (It might be stacked at local maxima) [39].

3.1.4 Inference Process

The inference process in the Bayesian network model often comes after the learning procedure. The inference process is related to inferring a collection of variables in a particular state, given the evidence about the state of other variables.

The *Probabilistic reasoning* in a Bayesian network model is the process of answers probabilistic queries based on new evidence [32]. In Bayesian statistics, the process of answers probabilistic queries, given new evidence, focuses on computing *posterior* probabilities. Thus, initially on the learned Bayesian network model, the beliefs represents initial probabilities or *prior* probabilities, before any evidence in the model; then, after new information about some variables is available, the updates beliefs represent the *posterior* probabilities. This procedure of updating the beliefs is defined as *probability propagation*. As discussed in 3.1.1, information flows in a Bayesian network model is not restricted to the defined orientation of the arrows between nodes; instead, the inference process can consider to *reasoning* in both bottom-top or top-bottom fashion [32].

Given a learned Bayesian network model: $\mathbf{B} = (\mathbf{G}, \mathbf{\Theta})$, and given new evidence $\mathbf{E}$ defined as an instantiation of one or more variables in the model, e.i.

$$\mathbf{E} = \{X_{i_1} = e_1, X_{i_2} = e_2, \ldots, X_{i_k} = e_k\}, \quad i_1, \ldots, i_k \in \{1, \ldots, n\} \tag{3.17}$$

we describe the behavior of the posterior probability distribution given previous information, by.

$$P(\mathbf{X}|\mathbf{E}) = P(\mathbf{X}|\mathbf{E}, \mathbf{B}) \tag{3.18}$$

Where $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. If we are interested in some subset of $\mathbf{X}$, say $\mathbf{Q} = \{X_{j_1}, X_{j_2}, \ldots, X_{j_l}\}$ with $j_1, \ldots, i_l \in \{1, \ldots, n\}$ and assuming that $\mathbf{Q}$ and $\mathbf{E}$ are disjoint, then the conditional probability query is:

$$P(\mathbf{Q}|\mathbf{E}) = P(\mathbf{Q}|\mathbf{E}, \mathbf{B}) \tag{3.19}$$

This probability represents the "*Marginal posterior probability distribution* of $\mathbf{Q}$" as say in [12], thus:

$$P(\mathbf{Q}|\mathbf{E}, \mathbf{B}) = \int P(\mathbf{X}|\mathbf{E}, \mathbf{B}) d(\mathbf{X} \setminus \mathbf{Q}) \tag{3.20}$$

The other type of queries is relating to computing the "configuration $\mathbf{q}^*$" from elements on $\mathbf{Q}$, such that, it has the maximum a posteriori queries, e.i.

$$MAP(\mathbf{Q}|\mathbf{E},\mathbf{B}) = \mathbf{q}^* = \underset{\mathbf{q}}{\operatorname{argmax}}\, P(\mathbf{Q}=\mathbf{q}|\mathbf{E},\mathbf{B}) \tag{3.21}$$

The maximum a posteriori queries and the conditional probability distributions establish a formal way to describe the inference process on the Bayesian network model. These inference mechanisms can be addressed in three different types of reasoning according to [32]:

Causal: This reasoning is when there exist recent information about causes. In agreement with the orientation of arcs between nodes described on the structure of the network, the probabilities in the effects can be updated.

Diagnostic: This reasoning is when performing diagnostic, e.i., reasoning from the effects in the direction to the possible causes. Technically, this reasoning is against the direction of the learned graph structure.

Intercausal: This reasoning way is when there exist causes with mutual nature, e.i., the causes with the same effect; this scenario usually describes the v-structures on the graph topology.

The inference process on a "large" Bayesian network model is a computationally-intensive problem. The worst-case scenario happens when; if there is an increment in the number of random variables, then the inference process problem will increase in an exponential computational complexity. Thus, the inference process in the Bayesian network models is an "NP-hard problem," as stated in [12]. However, in real-world applications, the inference process on Bayesian network models can efficiently be tackled by employing exact inference techniques over a "fair" number of variables. Moreover, we can use approximate inference techniques to handle more

variables in this process. In the following, we describe the two kinds of inference process algorithms.

Exact inference algorithms: In order to compute the exact values of $P(\mathbf{Q}|\mathbf{E},\mathbf{B})$, this class of algorithms merges iteratively local computations of Bayes rule. Nevertheless, the use of this class of algorithms is limited to a network with a few nodes or trees or multitrees structure. The two more popular algorithms in this inference scenario are the well-known *variable elimination* and *junction trees*.

- The *variable elimination* algorithm eliminates one by one of those variables, which are unnecessary for the probability query. The procedure uses the graph structure directly, defining the optimal sequence of operations on the local distributions and using dynamic programming design to save intermediate outcomes to avoid redundant calculations.
- The *junction trees* algorithm, transform the Bayesian network model into a junction tree, this transformation cluster the original nodes to reduce de network in a tree; the algorithm uses "Pearl's Message-Passing" procedure to compute probability queries [12].

Approximate inference algorithms: These algorithms simulate many samples from the learned Bayesian network model, then compute the conditional probabilities of interest, given the evidence, ponderating the samples that posses the evidence $\mathbf{E}$ and the query $\mathbf{Q} = \mathbf{q}$. Technically, the samples are extracted randomly and are usually known as *particles* in Machine Learning; for this reason, the algorithms based on samples are called *particle filters*. Algorithms were developed for random samplings, such as rejection sampling or importance sampling. The

sampling-based methods are diverse, from the simple process of simulate independent samples from local distribution to more sophisticated sampling known as *Markov Chain Monte Carlo* (MCMC) [12].

The inference process in a causal way will be described over the Bayesian network models. The causal interpretation of a probability query given some evidence is related to the direction of the arrows over the graph structure. Thus, the arrows can characterize the relationship in a casual way, rather than a probabilistic way. The queries in inference can be treated as probabilities of some causes of an event given its effects, or probabilities of some effects of an event given its causes. This process is known as *Causal Inference* [32].

The Bayesian network models cannot describe dependencies between variables in a temporal structure. The temporal characteristic of a problem arises when some feature evolution along time, and it is related to its past, even more, is related to other features over time. In many domains as medicine, finance, industry, etc., the temporal characteristic is relevant; thus, incorporate it on the process of modeling will be necessary. The Bayesian network model is extended to the temporal version to develop a precise probabilistic graphical model dynamically. This model is called a dynamic Bayesian network, which is described in the following.

3.2 Dynamic Bayesian Networks as Temporal Model

The *dynamic Bayesian network* model is a versatile probabilistic graphical model to characterize the stochastic processes compactly using a directed graphical model. The dynamic Bayesian network model (DBN) is a generalization of the well-known state-space model (SSM). Furthermore, the DBN model is a general case of the *Hidden Markov Model* (HMM), the *Kalman Filter Model* (KFM), and the *Vector Autoregressive* model (VAR). The DBN models allow to represent in a

compact factored form the underlying distribution of a temporal dataset; this representation is possible by using arbitrary probability distribution and applying a wide range of algorithms to perform learning and inference.

As in the static version, the objective of the learning process on a dynamic Bayesian network model is to discover a precise probabilistic graphical model of the underlying temporal distribution of the finite random sample dataset [40]. The versatility of the dynamic Bayesian network model allows representing a temporal dataset with a model graph structure along the time. The main reason for using a dynamic Bayesian network model is the capability to capture domain dynamic knowledge and reasoning under uncertainty supported on probability and graph theories. Note that the word "*dynamic*" represents in this context, modeling a "*dynamic system*," not changing the graph topology over time [10]. In the dynamic Bayesian network model, every random variable is described by many nodes along time; this situation does not happen in a static Bayesian network model. In the following, we present preliminary concepts to express a broad structure of the dynamic Bayesian network model.

Preliminaries: The dynamic Bayesian network model concerns probabilistic distributions of random attributes over time. Thus, usually, the terminology is to use capital letters "X, Y, Z" for random variables, lower case letters "x, y, z" to represent the instantiations from random variables, respectively. The set of random variables is usually described boldly as "$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$," consequently the instantiations version as "$\mathbf{x}, \mathbf{y}, \mathbf{z}$," respectively. The standard notation on a dynamic Bayesian network model is $\mathbf{X}[t] = \{X_1[t], \ldots, X_n[t]\}$ to represent the realization of a stochastic process of the dynamic model in a sorted way. $P(\mathbf{X}[t]) = P(X_1[t], \ldots, X_n[t])$ represent the joint probability distribution of the processes; thus, a dynamic Bayesian network model characterizes the probabilistic distribution of $\mathbf{X}[t]$. Finally, the time

is represented by t.

3.2.1 Dynamic Representation

A dynamic Bayesian network model extends the representation of a Bayesian network model described in 3.1.1 to propose a model for a temporal framework, assuming that changes occur between discrete-time slices. Consider the set of random variables $\mathbf{X}[t] = \{X_1[t], X_2[t], \ldots, X_n[t]\}$ in which the whole set represent a set of stochastic processes over time; each random process $X_i[t]$ with $i = 1, \ldots, n$ describes the random variable X_i at instant timestamp t. To represent probabilities about different trajectories of the set of stochastic processes $\mathbf{X}[t]$, the random variables and its distributions over $\mathbf{X}[0] \cup \mathbf{X}[1] \cup \ldots$ is needed to know; but, this distribution will be extremely complex. Assuming that $\mathbf{X}[t]$ is defined as a *Markovian first-order*; then, the following property holds the process:

$$P(\mathbf{X}[t+1]|\mathbf{X}[0], \ldots, \mathbf{X}[t]) = P(\mathbf{X}[t+1]|\mathbf{X}[t]) \tag{3.22}$$

Equation 3.22 can be interpreted as "The probability of the process $\mathbf{X}$ at instant time $t+1$ given all the past, only depends on the immediate past at time instant t". Another important assumption is that $\mathbf{X}[t]$ must be a *stationary process*; this is defined as $P(\mathbf{X}[t+1]|\mathbf{X}[t])$ does not depend on t, e.i., the probability distribution of the process does not change over time. If the previous restrictions hold, the dynamic Bayesian network model characterizes the joint probability distribution of a set of random variables and its possible trajectories from the stochastic processes $\mathbf{X}[t]$.

The dynamic Bayesian network model that we use in this research can be considered by two sub-models, as following:

- An initial Bayesian network model B_0, also known as prior model, which defines the prior distribution $P(\mathbf{X}[0])$, at timestamp $t = 0$.

- The transition Bayesian network model $B_{\rightarrow}$ which defines the transition distribution $P(\mathbf{X}[t+1]|\mathbf{X}[t])$ over the variables $\mathbf{X}[t] \cup \mathbf{X}[t+1]$, $\forall t$.

In the transition network $B_{\rightarrow}$, the probability distribution of the stochastic process, assuming the Markovian property will be:

$$P(\mathbf{X}[t+1]|\mathbf{X}[t]) = \prod_{i=1}^{n} P(X_i[t+1]|Pa(X_i[t+1])) \tag{3.23}$$

Where $X_i[t+1]$ represents the ith node in the graph model at timestamp $t+1$; $Pa(X_i[t+1])$ are the parents of $X_i[t+1]$ in the dynamic Bayesian network model. The observed version of equation 3.23 will be:

$$P(\mathbf{x}[t+1]|\mathbf{x}[t]) = \prod_{i=1}^{n} P(x_i[t+1]|Pa(X_i[t+1])) \tag{3.24}$$

Note that the transition network $B_{\rightarrow}$ is composed for a couple of time slice. The random variables defined on the first slice from $B_{\rightarrow}$ do not possess conditional parameters related to these. Instead, the variables in the subsequent slice, usually have parameters related with them; thus, there exist conditional probability distribution associated with these variables $P(X_i[t+1]|Pa(X_i[t+1]))$ with $t > 0$.
The arcs between consecutive time slices from "left to right" graphically depicts the *causal flow*. Inside each timestamp, links will represent instantaneous causalities. As a final remark, the structure of the initial network B_0 may be different from the intra-slice structure of the transition network $B_{\rightarrow}$ [10].

An example of a dynamic Bayesian network is displayed in Figure 3–3; both Bayesian networks, the initial and transition are displayed in different timestamps.

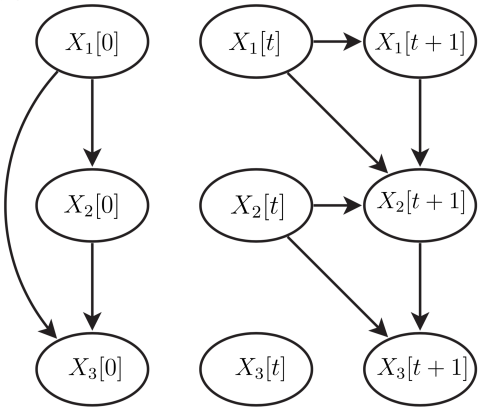

Figure 3–3: Initial and transition graphs characterizing a DBN for $X_1[t], X_2[t], X_3[t]$.

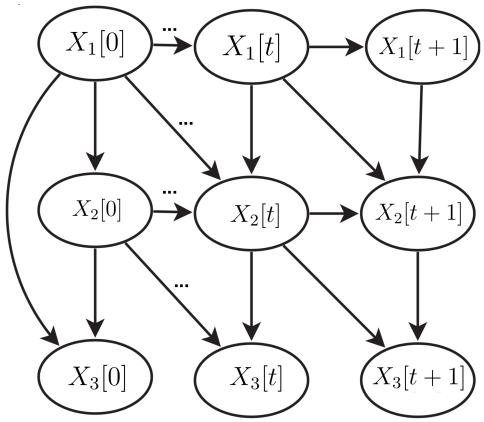

Figure 3–4: The corresponding "unrolled" network [2].

The initial graph B_0 represents a static Bayesian network model at the beginning of time $t = 0$. The transition graph $B_{\rightarrow}$ represents a transition, Bayesian network model. note that, in this network there exist a couple of vertical layers at time t and $t + 1$; in the initial layer, the arrows begin and finish in the other layer; moreover, no arrows from the finishing layer must go the beginning layer.

The dynamic Bayesian network model can be "unrolling," e.i., turning in just one full network. Figure 3–4 represents a dynamic Bayesian network model as an unrolled version on the network showed in Figure 3–3, corresponding to the same dynamic model. Note that in this unrolled version, the time-slice $t = 0$ corresponds to the initial graph B_0, and the subsequence time-slices corresponds to the transition graph $B_{\rightarrow}$.

Definition: A *dynamic Bayesian network* model is a pair of the form $(B_0, B_{\rightarrow})$, for describe the probability distributions of the stochastic processes defined on a collection of variables in a different timestamps $\mathbf{X}[0], \mathbf{X}[1], \ldots, \mathbf{X}[t], \ldots$.

The dynamic Bayesian network model $(B_0, B_{\rightarrow})$ is also known as 2-time slice Bayesian network model (2TBN). In real-life applications, the temporal dataset is defined on a finite time interval, e.i., $t = 0, 1, \ldots, T$; thus, the notation of the

dynamic Bayesian network model can be defined as unrolling the complete model structure in different timestamps; On timestamp $t = 0$, the parent nodes for $X_i[0]$ are defined as nodes into the initial graph B_0. On timestamps $t+1$, the parent nodes for $X_i[t+1]$ are in timestamps t or $t+1$; which correspond to the transition graph $B_{\rightarrow}$.

On the other hand, given a dynamic Bayesian network model, and using equation 3.23, the resulting *joint probability distribution* from the model, over the process $\mathbf{X}[0], \mathbf{X}[1], \ldots, \mathbf{X}[T]$, will be:

$$\begin{aligned} P(\mathbf{X}[0], \mathbf{X}[1], \ldots, \mathbf{X}[T]) &= P(\mathbf{X}[0]) \prod_{t=0}^{T-1} P(\mathbf{X}[t+1]|\mathbf{X}[t]) \\ &= \prod_{t=0}^{T} \prod_{i=1}^{n} P(X_i[t]|Pa(X_i[t])) \end{aligned} \tag{3.25}$$

Note that, $Pa(X_i[0])$ represents the ancestors of variable $X_i[0]$ in the initial graph B_0; $Pa(X_i[t])$ represents the parents of the nodes $X_i[t]$ in the transition graph $B_{\rightarrow}$, these parents usually are located on timestamps $t-1$ or t, for all $t > 0$.
The observed version of equation 3.25 is

$$\begin{aligned} P(\mathbf{x}[0], \mathbf{x}[1], \ldots, \mathbf{x}[T]) &= P(\mathbf{x}[0]) \prod_{t=0}^{T-1} P(\mathbf{x}[t+1]|\mathbf{x}[t]) \\ &= \prod_{t=0}^{T} \prod_{i=1}^{n} P(x_i[t]|Pa(X_i[t])) \end{aligned} \tag{3.26}$$

Both Equations 3.25 and 3.26 characterizes the uncertainty behavior of a dynamic Bayesian network model over a temporal dataset.

The versatility of the dynamic Bayesian network models allows handle stochastic processes like $Z[t] = (U[t], X[t], Y[t])$ for characterizing the entrance variable, unseen variable, and exit variable, respectively. These processes are frequently used in discrete-valued HMMs and their variants. Otherwise, the dynamic Bayesian network models handle not only first-order Markov property but naturally extends to

higher orders, e.i., kth order Markov process; thus, the 2TBN model will extend to a kTBN model [41]. Other forms of the dynamic Bayesian network models allow non-stationary stochastic processes; in this scenario, the topology of the graph continuously evolution on time [42]. So far, the dynamic Bayesian network models can handle continuous-valued hidden nodes like the KFMs. The transition probabilities are linear Gaussian distributions in KFMs; thus, there exists an injective relation among zeros on parameter and the nonexistence of arrows in the network. Another case of the dynamic Bayesian network model is the continuous-valued model VAR(p). In the VAR model, the conditional probability distributions are defined as Gaussian distribution. There exist an injective relation among zeros on the regression matrices of VAR(p) model and the nonexistence on the arrows defined on the inter-slice over a dynamic Bayesian network model. Finally, A hybrid DBN model with discrete and continuous nodes is the switching KFMs [10].

The learning process in a dynamic Bayesian network model from datasets is challenging; the central issue of learning is the nature of complex stochastic problem formulation, dynamically. For example, a few algorithms related to structure learning have been presented, using some concepts on the regular Bayesian networks, described in section 3.1.2; but, there exist differences between static and dynamic learning; some issues from dynamic learning scenario are:

- In order to describe the behavior of temporal sequences of large length, the parameters of the dynamic network model have to be linked between time-slices.
- The parameters related to $P(X_i[0])$ in the initial graph B_0, represents the first state into a dynamic scenario; thus, those parameters are usually determined separately from the transition graph $B_{\rightarrow}$.

- In order to estimate the graph structure of a dynamic Bayesian network model, the complete process must be divided in intra-slice and inter-slice structure connections. A static structure learning algorithms can learn the intra-slice structure connection for a Bayesian network model. The estimation of inter-slice structure connection will be identical to the problem of feature selection, because, in timestamp t, each variable have to possess its parents in the $t-1$ timestamp, assuming that the intra-slice relationship will be fixed. This assumption represents that dynamic structure learning is equivalent to the problem of variable selection.
  In the case when the dataset is complete, the application of well-known algorithms for feature selection can be applied without any restriction; those standard algorithms are forward, backward, or stepwise selection methods. However, if we have incomplete datasets, the process of structure learning will be computationally a bottleneck. An efficient alternative is the well-known "*Structural EM*" (SEM) algorithm described in [10].

The process of structure learning in the dynamic scenario usually extends methodologies from a static scenario. The structure learning provides and discovers conditional independencies on temporal datasets. In the following, we describe the dynamic structure learning.

3.2.2 Dynamic Structure Learning

Estimate the topology structure of a dynamic Bayesian network model from a temporal dataset, represent the extension of structure scoring rules for standard Bayesian network model in both complete and incomplete datasets. We now describe in detail the extension of the BIC-score structure learning described in equation 3.5, because this is the most relevant algorithm used in practice [2]. Since a dynamic Bayesian network model is a pair $(B_0, B_{\rightarrow})$, the main idea is to learn each Bayesian

network component by stages:

First, estimate the topology structure of the initial Bayesian network model B_0 like a regular structure learning process described in subsection 3.1.2, using the dataset defined over $\mathbf{X}[0]$.

Second, estimate the topology structure of the transition Bayesian network model $B_{\rightarrow}$, using the dataset defined over $\mathbf{X}[t] \cup \mathbf{X}[t+1]$.

If training data instances D (collection of values $\mathbf{x}[t]$) is available, corresponding to $\mathbf{X}[t]$, consisting of N_{seq} complete observation sequences, let l be the index for describing each sequence, e.i. $l = 1, \ldots, N_{seq}$. The lth sequence has length N_l with values $\mathbf{x}^l[0], \ldots, \mathbf{x}^l[N_l]$. Such a dataset gives us N_{seq} cases from initial timestamps; with this available information, the training process for initial model B_0 is possible. On the other hand, with $N = \sum_l N_l$ cases from the transition scenario, it is possible to train the transition model $B_{\rightarrow}$.

In order to introduce some notation, recall the subsection 3.1.3, we need an extension of equation 3.6. Let us define the parameters for the initial model B_0.

$$\theta_{ijk}^{(0)} = P(X_i[0] = j | Pa(X_i[0]) = k) \tag{3.27}$$

Where each variable $X_i[0]$ has r_i states; the number of configurations of $Pa(X_i[0])$ is equal to q_i, with $i = 1, \ldots, n; j = 1, \ldots, r_i; k = 1, \ldots, q_i$.

Similarly, define parameters for transition model $B_{\rightarrow}$.

$$\theta_{ijk}^{\rightarrow} = P(X_i[t] = j | Pa(X_i[t]) = k) \tag{3.28}$$

Where $t = 1, \ldots, T$. Let us consider an indicator function $I(\cdot; \mathbf{x}^l)$ equal 1 if the event "$\cdot$" happens over $\mathbf{x}^l$, and 0 in other cases. Now we can define sufficient statistics for initial model B_0

$$N_{ijk}^{(0)} = \sum_{l=1}^{N_{seq}} I(X_i[0] = j, Pa(X_i[0]) = k; \mathbf{x}^l) \tag{3.29}$$

and for transition model $B_{\rightarrow}$

$$N_{ijk}^{\rightarrow} = \sum_{l=1}^{N_{seq}} \sum_{t=1}^{T} I(X_i[t] = j, Pa(X_i[t]) = k; \mathbf{x}^l) \tag{3.30}$$

Assuming that G is a candidate dynamic Bayesian network structure, and using equation 3.25, then the likelihood function according to the structure of the dynamic Bayesian network model is:

$$\begin{aligned}
P(D|G) &= \prod_{t=0}^{T} \prod_{i=1}^{n} P(X_i[t]|Pa(X_i[t])) \\
&= \prod_{i=1}^{n} P(X_i[0]|Pa(X_i[0])) \times \cdots \times P(X_i[T]|Pa(X_i[T])) \\
&= \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{q_i} \theta_{ijk}^{(0)}, \times \cdots \times \theta_{ijk}^{\rightarrow} \\
&= \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{q_i} \left(\theta_{ijk}^{(0)}\right)^{N_{ijk}^{(0)}} \times \cdots \times \left(\theta_{ijk}^{\rightarrow}\right)^{N_{ijk}^{\rightarrow}} \\
&= \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{q_i} \left(\theta_{ijk}^{(0)}\right)^{N_{ijk}^{(0)}} \times \prod_{i=1}^{n} \prod_{j=1}^{r_i} \prod_{k=1}^{q_i} \left(\theta_{ijk}^{\rightarrow}\right)^{N_{ijk}^{\rightarrow}}
\end{aligned} \tag{3.31}$$

Then, the log-likelihood is given by:

$$l(G|D) = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{q_i} N_{ijk}^{(0)} \log\left(\theta_{ijk}^{(0)}\right) + \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{q_i} N_{ijk}^{\rightarrow} \log\left(\theta_{ijk}^{\rightarrow}\right) \tag{3.32}$$

Note that, the log-likelihood decomposes in parts that correspond to initial B_0 and transition $B_{\rightarrow}$ models; this facilitates the computation of BIC-score for the dynamic Bayesian network model, and it is given by:

$$BIC(G|D) = BIC_0 + BIC_{\rightarrow} \tag{3.33}$$

Where

$$BIC_0 = \sum_{i=1}^{n} \sum_{j=1}^{r_i} \sum_{k=1}^{q_i} N_{ijk}^{(0)} \log\left(\hat{\theta}_{ijk}^{(0)}\right) - \frac{d_0}{2} \log(N_{seq}) \tag{3.34}$$

And

$$BIC_{\rightarrow} = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{q_i} N_{ijk}^{\rightarrow} \log\left(\hat{\theta}_{ijk}^{\rightarrow}\right) - \frac{d_{\rightarrow}}{2}\log(N) \tag{3.35}$$

Where d_0 and $d_{\rightarrow}$ correspond the quantities of parameters on B_0 and $B_{\rightarrow}$, respectively. On the other hand, the penalty for BIC_0 depends on N_{seq}, whereas for $BIC_{\rightarrow}$ is on the total number of transitions observed N. The parameters estimate $\hat{\theta}_{ijk}^{(0)}$ and $\hat{\theta}_{ijk}^{\rightarrow}$ for B_0 and $B_{\rightarrow}$ respectively, are those who are optimizing log-likelihood 3.32 from dataset.

BIC-score has two important properties. First, the BIC-score has usually represented as a summation of terms, these terms will characterize the specific score from a selection of parents corresponding to specific random variables. A particular difference (add or remove an arrow) to a family will affect just the term on complete processes. The second property says that the term corresponding to calculate $X_i[t]$ given the information of its parent nodes, depends on the total counts ($N_{ijk}^{(0)}$ or $N_{ijk}^{\rightarrow}$) corresponding to its relatives. Furthermore, when retaining those total counts, it is possible to compute more families efficiently.

In order to find both graphs B_0 and $B_{\rightarrow}$, algorithms based on *hill-climbing* or *greedy* search exploit these two properties and progressively enhance a possible structure through the optimal way when the arrow can be added, removed, or changed the direction [43]. In the scenario of dynamic Bayesian network modeling, compared with the Bayesian network, it is usually to impose an additional restriction that consists of repeat the graph structure over time. This process decreases the search ways for each point; furthermore, the researching process for optimal graph structure for B_0 is frequently independent of the process of search the optimal graph structure for $B_{\rightarrow}$.

The structure learning process of a dynamic Bayesian network model based on scoring function, initially learn B_0, then $B_{\rightarrow}$; then, the user can choose a search algorithm that improves the learning process. In this line, the evolutionary and genetic

algorithms based on a greedy search mechanism was proposed in [44] for 2TBN structure learning, in which the process of learning the structure based on Bayesian optimization is performed in two phases. The initial phase concerns discovering the topology efficiently and compute the parameters related to the dynamic Bayesian network model; the next phase is related to obtain novel groups conformable with the discovered structure of a dynamic Bayesian network model. Using evolutionary algorithms but incorporating MCMC sampling methods for 2TBN structure learning is presented in [45].

An important variant in structure learning is proposed in [46]. It proposes a novel criterion to score the structure learning process in a dynamic Bayesian network model. This new criterion is based on the statistical concept of *cross-validation*; they state that the score generalizes efficiently compared with the BIC-score; then, their score is more appropriate than the BIC-score. In terms of efficiency, structure learning of the transition graph $B_{\rightarrow}$ is higher than the initial graph B_0; thus, in [47], the authors propose a "*Particle Swarm Optimization*" algorithm, to estimate the graph structure of a dynamic Bayesian network model. To learn the transition network $B_{\rightarrow}$ that duplicate the number vertices of B_0, it will use a stepwise concept, adding an arrow in the graph topology, and will reach to improve the learning structure problem.

The family of hybrid structure learning algorithms for a dynamic Bayesian network model, have a natural extension of static versions. The algorithms for estimating the topology in Bayesian networks based on a hybrid point of view, usually are called *Local Search Algorithms*. This kind of procedure tackles the problem by identifying the local structure network, then performs an optimization process on a candidate global model restricted to the available information locally. The process of identifying the local structure has the aim to discover the possible set of

candidates *Parent-Children* that represents the collection of target variables; One of this subroutine is the well-known "*Max-Min Parent Children*" algorithm [35]. A "state of the art" algorithm of this kind is the "*Max-Min Hill-Climbing*" algorithm (MMHC) proposed in [38], it combines local discovery "*Max-Min Parent Children*" subroutine and the well-known algorithm of search "*Greedy Search*" globally. Recently, the authors in [48], expanded an efficient procedure, named as "*Dynamic Max-Min Hill-Climbing*" (DMMHC); this procedure extends the well-known algorithm MMHC described in subsection 3.1.2, in order to take in account the dynamic scenario and used on a dynamic Bayesian network model.

3.2.3 Dynamic Parameter Learning

Methods for learning the parameters on a dynamic Bayesian network model are natural extensions of the static techniques over the regular Bayesian network models discussed in subsection 3.1.3; however, there are some slight differences between static and dynamic networks described above.

We already defined the parameters in equations 3.27 and 3.28 for B_0 and $B_{\rightarrow}$, respectively. Let D be the complete dataset, let G the graphical structure of the dynamic Bayesian network model already learned. Summing up the equation 3.31, we have the likelihood:

$$P(D|\Theta) = \prod_{i=1}^{n}\prod_{j=1}^{r_i}\prod_{k=1}^{q_i}\left(\theta_{ijk}^{(0)}\right)^{N_{ijk}^{(0)}} \times \prod_{i=1}^{n}\prod_{j=1}^{r_i}\prod_{k=1}^{q_i}\left(\theta_{ijk}^{\rightarrow}\right)^{N_{ijk}^{\rightarrow}} \tag{3.36}$$

Where $N_{ijk}^{(0)}$ and $N_{ijk}^{\rightarrow}$ are defined in equations 3.29 and 3.30 respectively. Then the log-likelihood is given by equation 3.32 and represented by

$$l(\Theta|D) = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{q_i} N_{ijk}^{(0)} \log\left(\theta_{ijk}^{(0)}\right) + \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{q_i} N_{ijk}^{\rightarrow} \log\left(\theta_{ijk}^{\rightarrow}\right) \tag{3.37}$$

The aim is to optimize the log-likelihood with respect to Θ, e.i., find $\arg\max_{\Theta} l(\Theta|D)$ for finding the MLE for parameters $\theta_{ijk}^{(0)}$ and $\theta_{ijk}^{\rightarrow}$.

The log-likelihood in 3.37 represents a sum of two quantities, both quantities rely on conditional probabilities of random features, given its ancestors; in order to find the MLE is necessary to optimize within every family independently, this means that, learning the parameter $\theta^{(0)}_{ijk}$ independently of $\theta^{\rightarrow}_{ijk}$. Thus, using the standard maximum likelihood estimate, we immediately get the following expressions for $\hat{\Theta}$, first for the MLE in B_0:

$$\hat{\theta}^{(0)}_{ijk} = \frac{N^{(0)}_{ijk}}{\sum_j N^{(0)}_{ijk}} \tag{3.38}$$

Then, for the MLE in $B_{\rightarrow}$:

$$\hat{\theta}^{\rightarrow}_{ijk} = \frac{N^{\rightarrow}_{ijk}}{\sum_j N^{\rightarrow}_{ijk}} \tag{3.39}$$

On the other hand, a straightforward extension in the case of Bayesian estimation for Θ is presented using conjugate Dirichlet prior. To obtain a "closed-form solution," the decomposition of the prior will be

$$P(\Theta) = \prod_{i,k} P(\theta^{(0)}_{i.k}) \times \prod_{i,k} P(\theta^{\rightarrow}_{i.k}) \tag{3.40}$$

This factorization is possible if the prior over each conditional probability is independent of others. The specific distribution Dirichlet prior, is required to a conjugate Bayesian analysis for a multinomial distribution. The hyperparameters for the prior are $\{N'_x : x \in Val(X)\}$ in both networks B_0 and $B_{\rightarrow}$, then:

$$P(\theta^{(0)}_{i.k}) \propto \prod_j \left(\theta^{(0)}_{ijk}\right)^{N'_x - 1} \tag{3.41}$$

and

$$P(\theta^{\rightarrow}_{i.k}) \propto \prod_j \left(\theta^{\rightarrow}_{ijk}\right)^{N'_x - 1} \tag{3.42}$$

Thus the prior distribution will be.

$$P(\Theta) \propto \prod_{i,k} \prod_j \left(\theta^{(0)}_{ijk}\right)^{N'_x - 1} \times \prod_{i,k} \prod_j \left(\theta^{\rightarrow}_{ijk}\right)^{N'_x - 1} \tag{3.43}$$

From equations 3.36 and 3.43 we obtain the posterior distributions for Θ.

$$P(\Theta|D) \propto \prod_{i,k} \prod_{j} \left(\theta_{ijk}^{(0)}\right)^{N_{ijk}^{(0)}+N_x^{'}-1} \times \prod_{i,k} \prod_{j} \left(\theta_{ijk}^{\rightarrow}\right)^{N_{ijk}^{\rightarrow}+N_x^{'}-1} \tag{3.44}$$

Since we have each variable $X_i[0]$ with r_i states, the number of configurations of $Pa(X_i[0])$ is equal to q_i, with $i = 1, \ldots, n; j = 1, \ldots, r_i; k = 1, \ldots, q_i$.
Similarly for $X_i[t]$ with $t > 0$. Then each hyperparameter $N_x^{'}$ has the same configurations, e.i., $N_x^{'} = N_{ijk}^{\prime(0)}$ in B_0 and $N_x^{'} = N_{ijk}^{\prime\rightarrow}$ in $B_{\rightarrow}$. Thus the Bayesian maximum a posteriori (MAP) estimates of Θ, such that $\Theta^* = \arg\max_{\Theta} P(\Theta|D)$ are:

$$\hat{\theta}_{ijk}^{(0)} = \frac{N_{ijk}^{(0)} + N_{ijk}^{\prime(0)}}{\sum_j \left(N_{ijk}^{(0)} + N_{ijk}^{\prime(0)}\right)} \tag{3.45}$$

and

$$\hat{\theta}_{ijk}^{\rightarrow} = \frac{N_{ijk}^{\rightarrow} + N_{ijk}^{\prime\rightarrow}}{\sum_j \left(N_{ijk}^{\rightarrow} + N_{ijk}^{\prime\rightarrow}\right)} \tag{3.46}$$

Parameter learning from incomplete datasets has the main difficulty that can no longer decompose as in equation 3.36. This scenario can be explained, like the optimal parameter selected on a piece of the graph depends on the selection of the parameter on another piece of the graph [2]. As in the static scenario, the EM procedure is used. The expectation process will complete the dataset by estimating the expected total counts, using the present computed parameters. Then, the maximization process optimizes the likelihood again for an estimate of the parameters to maximize the likelihood distribution; this process estimates the parameters like if expected total counts were counts, which observed correctly.

3.2.4 Dynamic Inference Process

The process of dynamic inference is related to infer some specific status of a subset of random features given few preliminary information or evidence of the state

of other variables, on a specific timestamp. Thus the goal of inference is to calculate probabilities of the form $P(X_i[t]|x_{1:\tau})$ defined as marginals, where $X_i[t]$ represents the i-th random variable at time-instant t and $x_{1:\tau}$ represents the evidence.
There exist three cases in the process of inference:
When $\tau = t$, the inference process is known as filtering (also called tracking), here the process is to compute probability queries of some variables in the actual instant time on the dynamic network, given the complete accessible information.
When $\tau > t$, the inference process is named smoothing, here the aim is to remove the noise from temporal data over the past of the data, given the current information of the data until the present timestamp.
When $\tau < t$, the inference process is known as forecasting or prediction.

Computationally, the inference process in the dynamic Bayesian network models is defined as an *NP-hard problem*. Usually, the approaches to solve the inference process is divided into a couple of categories: the exact and approximate inference, in both inference in a static Bayesian network model, is called as subroutines.

- In exact inference, the *Junction tree* algorithm is the most popular. It decomposes the computations of joint probability into a linked set of local computations by transforming a dynamic Bayesian network structure into a *Clique tree*. The sum-product algorithm is applied to compute the probability queries of interest. Other popular algorithms presented in the literature are the smoothing forward-backward, frontier algorithm which consist of sweep a Markov blanket, then perform forward-backward on frontier collection F, over the dynamic Bayesian network model [10]. The computational problems with exact inferences force to use approximate inference.

- Approximate inference algorithms are deterministic or stochastic.

  Deterministic approaches are based on variational inference techniques, approximating the target probabilities with other analytically. First, picking a family of distributions with specific parameters, then the parameters are varied such that the approximation is close to target. Finally, it is used as probabilistic queries of interest [12]. The most common algorithms are the *Boyen-Koller* algorithm, which computes the joint probability distribution approximately using an alternative like multiplication of marginal distributions. The *factored frontier* algorithm that describes the boundary distribution represented on factored model, and the generalization of both previous algorithms, the *loopy belief propagation* algorithm is described in detail in [10].

  Stochastic approaches are based on numerical sampling, usually knowns as *Monte Carlo* techniques. The idea is to approximate the probability distribution with samples to obtain probability queries; the sampling procedures are based on *Markov Chain Monte Carlo* methods (*Metropolis-Hastings, Gibbs sampling, Slice sampling, Simulated annealing*) or the *Importance sampling* algorithms [10]. Another important algorithm in an online fashion is the well-known *Particle Filtering*, a kind of Importance Sampling method in a sequential mode. A drawback of the sampling procedure is the computational speed; usually, the time speed is slow compared with deterministic algorithms.

Hybrids algorithms that combine exact and stochastic inference are proposed in the literature. The concept behind is to combine exact and approximate scenarios, some features adopting from exact algorithms, and the remain features adopting the approximate algorithms based on methods of sampling. This complete hybrid procedure is defined as the *Rao-Blackwellisation*; this procedure is based on the theorem of *Rao-Blackwell*, which states the way to make better an estimator subject to each convex loss function [12].

Specifically, when using both procedures *particle filtering* and *Rao-Blackwellisation*, the technique is known in the literature as the *Rao-Blackwellised Particle Filtering*. As a final remark of this section, in this research work is not relevant to use inference algorithms.

The objective of this research work is to discover meaningful temporal anomalies and provide an explanation using a dynamic Bayesian network model and probabilistic association rules. The described fundamental concepts, methods, and algorithms about the dynamic Bayesian network models represent the foundations to describe the domain knowledge in data. In the following, we will describe concepts about probabilistic association rules as a complementary method to reach our objectives. Finally, the discretization method will be necessary in order to delimit our scope.

3.3 Probabilistic Association Rules

The association analysis has an objective, which is to discover "interesting" relationships between items. The association rules are patterns to discover "interesting" relations between variables in datasets.

The *interestingness measures* have an aim to discover and rank patterns (association rules) corresponding to the interest of the researcher. The concept of "interestingness" is a whole branch in Data Mining, it has foundations on nine principles: "*emphasizes, conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility, and actionability.*" These principles are widely accepted in specific situations in the knowledge discovery process and data mining tasks. Usually, the principles have been used to determine if a discovered pattern is interesting or not [14].

The nine principles are organized on *objective, subjective, and semantic* measures. The measures to reach "objectiveness" are based on three fundamentals: The datasets

as evidence, none previous knowledge is required, and foundations on probability theory. Thus, the "interestingness" and "objectiveness" measures will be based on probability. Finally, we can provide an intuitive concept of probabilistic association rule as an "Interesting and meaningful discovered pattern."

Definition: An association rule is a conditional statement of form $X \longrightarrow Y$, where X and Y represent a disjoint collection of objects.

In our scenario, an association rule can be defined as a conditional statement between random variables in a dataset.

In order to select "interesting association rules" from the complete collection of rules, there exist a couple of objective measures that will be used:

Support: "$supp(X) = n_X/n$," where X is an itemset. The support is the rate of sharing of "transactions in a database that contains X." The probabilistic interpretation of support corresponds to estimate $P(X) = supp(X) = n_X/n$, the "prior probability of the itemset X is contained in a transaction."

Confidence: "$conf(X \longrightarrow Y) = supp(X \cup Y)/supp(X)$" represent confidence of "$X \longrightarrow Y$." "The confidence measures the proportion of sharing of transactions containing Y in all the transactions containing X." The probabilistic interpretation of confidence correspond to calculate conditional probability "$P(X|Y) = conf(X \longrightarrow Y) = P(X \cap Y)/P(X)$."

With both objective measures, the definition of a probabilistic association rule is:

Definition: "Given that:

$P(X \longrightarrow Y) = P[supp(X) > minsupp \wedge conf(X \longrightarrow Y) \geq minconf]$.

If $P(X \longrightarrow Y) \geq minprob$, then $X \longrightarrow Y$ is a probabilistic association rule."

The parameter *minprob* is defined as a probability threshold; the parameters

minsupp and *minconf* are referred to as minimum support and confidence, respectively.

Note that the parameters *min* can be replaced by *max* in order to produce other probabilistic association rules. We applied two pre-defined probabilistic association rules with specific parameters. We applied two pre-defined probabilistic association rules with specific parameters. Finally, using these probabilistic association rules based on a dynamic Bayesian network model. It will be possible to discover a domain specific temporal anomalous patterns to determine interesting temporal outliers provided by its contextualization.

3.4 Discretization

In this research work, we use discrete dynamic Bayesian network models, which is a network with discrete-valued stochastic processes. The efficient algorithms for structure and parameter learning that have been proposed are dedicated to discrete random variables. Instead, in the continuous case, the dynamic structure learning algorithms are very scarce, and as far as we know, does not exist an efficient algorithm for this scenario. Thus, a data discretization will be required to improve the performance in the discovery process. Generally, the discrete representation allows a tractable computational analysis [49]. Thus, there is a significant reduction in the computational cost in the dynamic structure learning algorithms for dynamic Bayesian networks.

Time series are realizations of a continuous stochastic process. The time series discretization consists of transforming it into a discrete sequence. This process must preserve the relevant relationship within and between random processes. Most of the discretization methods are unsupervised; these methods are used in situations when the labels on the dataset are not available [50]. Since this thesis tries to uncover temporal outliers, a necessary and sufficient condition is to perform an equal- width

discretization method, because our objective is to discover outliers. Otherwise, if we use a very sophisticated and robust discretization method, we lose information about the outliers, and in advance will be not possible to discover those outliers.

The discretization transform time series: $X[t] = \{x[1], x[2], \ldots, x[n]\}$ into a discrete sequence $Y[t] = \{y[1], y[2], \ldots, y[n]\}$. Discretization is performed recursively on an attribute, selecting thresholds to divide the range of the variable $X[t]$ in equal-width intervals or binds, thus the range of the variable $Y[t]$ will be $1, 2, \ldots, |bins|$. The number of bins ($|bins|$) was decided under an experimental study, according to sensitivity analysis in the discovery process of interesting temporal outliers.

Chapter 4 DETECTING INTERESTING TEMPORAL OUTLIERS

4.1 Introduction

Temporal outlier analysis has the main objectives to discover and analyze the *Temporal anomalous patterns* in structured temporal datasets. As in the static version, temporal outlier detection does not only aim to find outliers but also to provide explainability of the reported anomalies, and to associate anomalies with physical scenarios in order to enhance the domain knowledge. Generally, to provide explainability, it is necessary to find physical scenarios and represent them as a research problem. One way to represent physical scenarios or physical events is to represent it by subspaces (a subset of random variables) over the collection of stochastic processes under the research problem. These subspaces of variables into the dynamic Bayesian network model will represent contextualizations and will provide explainability of the discovered temporal anomalies. The main contribution of this research work goes in line to discover interesting temporal outliers and provide their explainability.

As an example of the problem to provide explainability and find physical events of the reported temporal outliers, we describe a particular case in the following:

The abrupt change in the stock market in a particular period, known as "The flash crash of May 6, 2010," was a temporal anomalous pattern of the stock market in the USA. This temporal anomaly should have been discovered in advance since it should have been related to anomalous physical events in order to provide explainability about what were the possible causes. In this context, examples of anomalous

physical events will be economic factors like the rare event registered previously in the stock market known as the "toxic order imbalance." These orders were high and were registered in a short period previous to the collapse. Another physical rare temporal pattern was "the inadvertent large sell order for Procter & Gamble stock," encouraging huge trading orders by financial algorithms; thus, these events could have caused the crash in the stock.

Those two explained dynamic physical anomalous events related to "the flash crash of May 6, 2010," are examples of domain knowledge of an expert. However, the capability of a dynamic Bayesian network model allows us to capture this specific knowledge with a suitable database. Thus, we can tackle the issue of associating a temporal anomaly with the anomalous physical event to provide explainability, by making use of the qualities of a dynamic Bayesian network model.

On the other hand, is well known that in Data Mining community, if "A" represents the collection of uncovered temporal anomalies from the dataset by a method over a specific research problem, and "B" represent the "unknown true" collection of temporal anomalies over the same research problem. Then, the "ideal" temporal anomaly detection algorithm, should have a high statistical performance measure: "Precision $P(B|A)$, and Recall $P(A|B)$." However, in general, access to the collection of true anomalies "B" is impossible; then, temporal outlier detection algorithms must be done in an unsupervised mode like clustering techniques. Our approach using *Dynamic Bayesian Networks and Probabilistic Association Rules* has the advantage of discovering temporal anomalies in unsupervised mode.

As far as we know, the "state of the art" methods for detecting temporal outliers in multidimensional datasets, are based on the reduction of dimensionality, losing valuable information not only in the behavior of the original dataset but also in the causes of the rare events that are related with outliers. The techniques based on

dimensionality reduction, generally apply algorithms to detect anomalies in a univariate temporal sequence, without considering another kind of relevant information like crosscorrelation or dependency relation between temporal sequences. These two mentioned drawbacks make the recent techniques inefficient in order to detect interesting temporal outliers, much less associate them with rare events and explain the circumstances where and why they happened.

Instead, the dynamic Bayesian network model, despite being computationally intensive, it can work with complete multidimensional data. Also, it describes and captures the underlying domain knowledge of the dataset. Then resorting to probabilistic association rules, it is possible to mine anomalous patterns and subspaces when they happen, to finally discover "interesting" or "real" temporal outliers.

A property of all temporal anomaly detection techniques is that time represents an essential characteristic in the way of formulating the "*Temporal anomalous patterns*" in order to detect them as outliers. A main peculiarity of *time* is the ordered nature of the temporal dataset since the historical evolution of one variable depends intrinsically on its past. Moreover the evolution of one variable can depend on the evolution of other variables over time. A dynamic Bayesian network model well represents this characteristic of dependency since this model represents naturally complex stochastic processes. Furthermore, *time* represents a natural contextualization of temporal datasets, since a "normal" observation or sequence can occur in an instant time "t_1" or time window $[t_i, t_{i+w}]$ with a specific value. However, another observation or sequence with the same value as the previous one can occur in other instant time "t_2" or another time window $[t_j, t_{j+w}]$ can be considered as "unusual." This phenomenon can happen because of the behavior of the natural components of a stochastic process like trends, seasonalities, cycles, or noises. Typically, the dynamic Bayesian network model can describe and fit the components of a stochastic process and allow us to handle time contextualization.

Intuitively we have a multidimensional dataset composed of temporal discrete sequences, and time series with equal or different sizes (length) but with same time granularity, e.g., days, months, etc. Suppose we place time sliding windows with an appropriate width along the time, these windows may be overlap or not. Over the windows, each temporal sequence follows a global regular stochastic process, e.g., stationary process, Markovian process, etc., into each temporal sequence there exist relevant properties, as a degree of autocorrelation with its past, and a degree of cross-correlation between temporal sequences.

Traditionally, temporal anomalies will represent points or sequences over temporal datasets. The temporal anomalies usually are found on sparse regions into a specified time-window. Moreover, temporal anomalies do not follow the behavior of an appropriate stochastic process. On the foundations of *Temporal Pattern Mining*, the sparse regions usually are defined as regions with *low support*, then, these regions contain the temporal outliers; moreover, those mined outliers go against or disrupt the normal behavior of the stochastic process model.

In order to discover temporal anomalous patterns, and associate them with physical anomalous events represented by subspaces of variables, this research work shows that there exists a necessity to focus on the well-known concept as *confidence*, to yield more meaningful results, since confidence gives us information about the conditional probability of an event given prior information. In this context, prior information refers to the uncertainty states of parent nodes of a particular node, and conditional information refers to the uncertainty states of the child nodes given its parents. Both prior and conditional information are defined in an instant time over a dynamic Bayesian network model.

4.1.1 Problem Statement and Contribution

We propose here a new approach to discover and explain temporal anomalies, which couple two fundamental frameworks in data mining, the *dynamic Bayesian networks* models, and the *probabilistic association rules*, inspired and adapted from the seminal work on "*Inferring anomalies from data using Bayesian networks*" proposed by Babbar, 2013 [13]. As far as we know, the research mentioned above is widely considered a contemporary and unique method of this kind. We are convinced that these approaches and methods to detect "real" and "interesting" temporal outliers can suitably extend dynamically and bear in mind the temporal dimension in datasets.

The dynamic Bayesian network model has the capability of exploring the causality and probabilistic dependency in the feature space from stochastic processes, through algorithms for dynamic structure learning, consider the degree of autocorrelation and crosscorrelation. Moreover, the dynamic Bayesian network model can capture and organize the uncertainty information about the domain knowledge underlying in a temporal dataset through the algorithms for dynamic parameter learning.

On the other hand, the two probabilistic association rules, set up as:

- "*Low Support & High Confidence.*"
- "*High Support & Low Confidence.*"

Will allow us to find scenarios where the discrepancies between prior and conditional probabilities are statistically significant. It is necessary to specify that we are not looking for frequent patterns like in traditional association mining rules; instead, the aim is to discover unusual patterns, whose existence on the temporal dataset will be uncommon events and exceptional along the time. Finally, when we found these discrepancy scenarios, we can discover temporal anomalous patterns, and report the

explainability of the "interesting" and "real" temporal outliers within a specific domain knowledge.

From now, we will denote the uncovered temporal anomaly pattern as "*Domain Specific Temporal Anomalous Pattern*" (DSTAP). One DSTAP will represent a subspace of random variables over the stochastic process under the specific research problem. Furthermore, a DSTAP is a substructure (subgraph) inside of the whole graph structure related to the dynamic Bayesian network model, within a specific time window, where at least one of the two aforementioned probabilistic association rules will fulfill. In the test phase, a particular observation or a subsequence is declared as a temporal anomaly by the method into a specific domain, if and only if, this temporal anomaly will belong to one of the discovered DSTAP. Note that, we will have a set of discovered DSTAP, along the observation time. On ahead of this chapter, we address the problem statement summary as:

Input: Given a collection of the temporal dataset.

Output: Discover and explain interesting temporal outliers over a specific domain.

We reach our objectives through the main contribution of this thesis. We are providing a contemporary framework that coupled two methods, the *dynamic Bayesian network* model and *probabilistic association rules* with the primary objective to uncover temporal outliers in datasets. Based on *causality* foundations to provide characterization and description of why an item is declared as an anomaly. The proposed method is specially designed to yield a *contextual scenario* of the outlier, which is a piece of valuable information that can be used to enhance the knowledge efficiently from the temporal anomalies.

4.2 Methodology: Domain Specific Temporal Anomalous Patterns

This section is split into two parts. The first part is dedicated to structure and parameter learning of a dynamic Bayesian network model from the temporal dataset. The second part is about the details of the two probabilistic association rules, and how they use to discover temporal anomalous patterns.

4.2.1 Learning a Dynamic Bayesian Network Model From Dataset

Learning the network topology and parameters of a dynamic Bayesian network model from the dataset is conducted on local search algorithms for structure learning described in subsection 3.2.2, and maximum likelihood estimation for parameter learning explained in subsection 3.2.3.

Consider a dynamic Bayesian network model $(B_0, B_{\rightarrow})$. Learning the initial and transition graphs are performed independently. In both networks, the learning structure is done in a "static" mode; first learn structure and parameters for B_0 with dataset $\mathbf{X}[t = 0]$, then for $B_{\rightarrow}$ with dataset $\mathbf{X}[t] \cup \mathbf{X}[t + 1]$.

The general approach of learning the structure of a dynamic Bayesian network is made in two phases: First, identify a local structure set of "Parent-Children." Second, perform an optimization of the global model, constrained to previous local information. An algorithm to discover the candidates "Parent-Children" is proposed in [51] as a Dynamic Max-Min Parent Children (DMMPC); this algorithm is composed of two sub-procedures, the neighborhood identification and symmetrical correction.

The neighborhood identification algorithm is denoted as $\overline{\text{DMMPC}}$; this identifies the neighborhood Ne_0 of the target node T in B_0 and the neighborhood Ne_+

of the T in $B_{\rightarrow}$.

In the initial network B_0, Ne_0 is part of $\mathbf{X}[0]$ and $\mathbf{X}[1]$, thus T can have parents or children in $t = 0$, and only have children in $t = 1$. Then $\overline{\text{DMMPC}}$ uses the static version algorithm MMPC proposed in [38]. Finding the neighborhoods are based on the Maximum and Minimum heuristic procedure, which returns the maximum overall variables of the minimum association with T relative to Ne_0, using an association measure like χ^2 to represent the degree of dependency between nodes.

In the transition network $B_{\rightarrow}$, Ne_+ is part of $\mathbf{X}[t-1]$, $\mathbf{X}[t]$, and $\mathbf{X}[t+1]$; thus, T can have parents or children in time t, only parents in time $t-1$, and only children in time $t+1$. Similarly, $\overline{\text{DMMPC}}$, perform the static version algorithm MMPC. Note that the orientations of arrows between time slices are from $t-1$ to t, or t to $t+1$.

The symmetrical correction is performed in the if line. A node X belongs to the neighborhood of T; the opposite is also true. In a dynamic scenario, the symmetrical correction is due to the non-symmetry of temporality.

In the initial network B_0, Ne_0 is divided in the set of parents or children of T and the set of children of T, in both sets the correction " If a node X belongs to the neighborhood of T, the opposite is also true" must be fulfilled, if not X is removed from neighborhood.

In the transition network $B_{\rightarrow}$, Ne_+ is divided into the set of parents of T in time $t-1$, the set of parents or children of T in time t, and the set of children of T in time $t+1$. The symmetrical correction is applied to the neighborhood of T.

Once the local structure identification of "Parent-Children" is made, the optimization process is needed. The dynamic structure learning algorithm identifies the graphs B_0 and $B_{\rightarrow}$ independently, through an adaptation of the greedy search algorithm proposed in [2], constrained with the local information available by the

identification of Ne_0 of each node in B_0, and Ne_+ of each node in $B_{\rightarrow}$. Thus, the complete procedure to learn the dynamic structure is called "Dynamic Max-Min Hill-Climbing" (DMMHC). This algorithm adds an edge during the greedy search, if and only if the starting node is in the neighborhood of the ending node. In the case of B_0, add an edge is perform only to nodes in the set of parents or children of another node in $t = 0$. Instead, in $B_{\rightarrow}$; adding edges with the constraints of inter-dependencies between $t - 1$ and t; and intra-dependency in t. Then, we can describe the dependency structure of the temporal data by learning the topology of the dynamic Bayesian network model $(B_0, B_{\rightarrow})$. In the following, we show the main algorithm DMMHC, then the subroutines that are recursively called.

Algorithm 1, estimates the dynamic structure of a dynamic Bayesian network from the dataset. First, learning the initial network B_0, then the transition network $B_{\rightarrow}$ independently. This algorithm calls a subroutine to find the set "Parents-Children" and performs a greedy search.

Algorithm 2, represent a subroutine of Algorithm 1, and performs the identification of the neighborhoods of a node (Ne_0 and Ne_+), and the symmetrical correction of each node (" If a node X belongs to the neighborhood of T, the opposite is also true"); this algorithm calls another, the Algorithm 3 which mostly finds the neighborhoods of a target node.

Algorithm 1 DMMHC.

Input: Dateset D.

Output: DBN=(B_0, $B_{\rightarrow}$).

% Initial graph B_0

for all $X \in \mathbf{X}[t=0]$ **do**
  CPC_X=DMMPC$(X, D).CPC_0$ {CPC_0= set of parents or children in $t=0$}
  CC_X=DMMPC$(X, D).CC_1$ {CC_1= set of children in $t=1$}
end for
if $Y \in CPC_X$ **then**
  add.edge $Y \rightarrow X$
end if

% Transition graph $B_{\rightarrow}$

for all $X \in \mathbf{X}[t]$ **do**
  CPC_X=DMMPC$(X, D).CPC$ {CPC= set of parents or children in t}
  CC_X=DMMPC$(X, D).CC$ {CC= set of children in $t+1$}
  CP_X=DMMPC$(X, D).CP$ {CP= set of parents in $t-1$}
end for
if $Y \in CPC_X$ and $X, Y \in \mathbf{X}[t]$ **then**
  add.edge $Y \rightarrow X$
else if $Y \in CC_X$ and $X \in \mathbf{X}[t]$ and $Y \in \mathbf{X}[t+1]$ **then**
  add.edge $X \rightarrow Y$, and Do not *reverse.edge* $X \rightarrow Y$
end if

Algorithm 3, identify the neighborhoods of a target node. It uses as a subroutine the static algorithm of local search, which is described in Algorithm 4.

Algorithm 4, identifies neighborhoods of a node in a non-temporal dataset, this procedure uses as a dependency measure an association function $Assoc$, that in our case is based on a χ^2 measure.

According to [38], the time complexity of Algorithm 4 is governed by the dependency test for the target node with other nodes conditioned on the sets of parents or children (Ne) in the dataset. Consider the dataset with "n" nodes in the Bayesian network, consider $|Ne|$ the number of nodes in the set of parents or children; then the time complexity is bound by $O(n2^{|Ne|})$. The temporal extension of structure learning done in [51] establishes that in Algorithm 3, there exist two cases, when $t=0$ and $t>0$. In the first case, it computes an association measure of all nodes

Algorithm 2 DMMPC.

Input: Target node T, Dateset D.
Output: Ne_0, Ne_+

$Ne_0 = \overline{\text{DMMPC}}(T, D).Ne_0$
$Ne_+ = \overline{\text{DMMPC}}(T, D).Ne_+$
$CPC_0 = Ne_0 \cap \mathbf{X}[0]$
$CC_1 = Ne_0 \cap \mathbf{X}[1]$
for all $X \in CPC_0$ **do**
  if $T \notin \overline{\text{DMMPC}}(X, D).Ne_0$ **then**
    $CPC_0 = CPC_0 \setminus \{X\}$
  end if
end for
for all $X \in CC_1$ **do**
  if $T \notin \overline{\text{DMMPC}}(X, D).Ne_+$ **then**
    $CC_1 = CC_1 \setminus \{X\}$
  end if
end for
$Ne_0 = CPC_0 \cup CC_1$
for all $X \in Ne_+$ **do**
  if $T \notin \overline{\text{DMMPC}}(X, D).Ne_+$ **then**
    $Ne_+ = Ne_+ \setminus \{X\}$
  end if
end for
$CPC = Ne_+ \cap \mathbf{X}[t]$
$CC = Ne_+ \cap \mathbf{X}[t+1]$
$CP = Ne_+ \cap \mathbf{X}[t-1]$

Algorithm 3 $\overline{\text{DMMPC}}$.

Input: Target node T, Dateset D.
Output: Ne_0, Ne_+

$ListC_0 = \mathbf{X}[0] \setminus \{T\} \cup \mathbf{X}[1]$
$Ne_0 = \overline{\text{MMPC}}(T, D, ListC_0)$
$ListC = \mathbf{X}[t-1] \cup \mathbf{X}[t] \setminus \{T\} \cup \mathbf{X}[t+1]$
$Ne_+ = \overline{\text{MMPC}}(T, D, ListC)$

Algorithm 4 $\overline{\text{MMPC}}$.

Input: Target node T, Dateset D, List of candidates $ListC$.
Output: Neighborhood of T (Ne), set of parents or children.

$Ne = \emptyset$
repeat
  $assocF = \max_{X \in ListC} \min_{S \subseteq Ne} Assoc(X; T|S)$
  $F = \arg\max_{X \in ListC} \min_{S \subseteq Ne} Assoc(X; T|S)$
  if $assocF \neq 0$ **then**
    $Ne = Ne \cup \{F\}$
    $ListC = ListC \setminus \{F\}$
  end if
until Ne has no change or $assocF = 0$ or $ListC = \emptyset$
for all $X \in Ne$ **do**
  if $\exists S \subseteq Ne$ and $Assoc(X; T|S) = 0$ **then**
    $Ne \setminus \{X\}$
  end if
end for

(in $t = 0$ and $t = 1$), with the target node in $t = 0$, conditioned on the set Ne_0; assuming that n represents the number of nodes, then the time complexity is bounded by $O(2n2^{|Ne_0|})$. In the second case, there exist three timestamps $t-1, t$, and $t+1$, in which the target node could be associated, conditioned with set Ne_+, then the time complexity is bounded by $O(3n2^{|Ne_+|})$. Algorithm 2, recall the Algorithm 3 two times sequentially, and performs the symmetrical correction in each loop and bounded by the total number of nodes n. Thus, summing up the computational cost in Algorithm 2, the time complexity is bounded by $O(3n2^{|Ne_+|})$. Finally, the total computational cost of construct the graph topology of a dynamic Bayesian network model $(B_0, B_\rightarrow)$, described in Algorithm 1, is bounded by $O(|3n|^2 2^{|Ne|}) \approx O(n^2 2^{|Ne|})$. Because Algorithm 2 is recalled into a loop related to the nodes in the graph in $t = 0$ and $t > 0$. Note that Ne represents the largest set of the neighborhood in $t-1, t$, and $t+1$ over all nodes in the time t.

Once the graph structure of the dynamic Bayesian network model is made, the next step is to perform parameter learning of the model. The procedure to estimate the parameters of the probability distribution of each node in the discrete case is well

explained in subsection 3.2.3 through the procedure of maximum likelihood.

4.2.2 Two Probabilistic Association Rules

The learned dynamic Bayesian network model $(B_0, B_{\rightarrow})$, describes the probabilistic relation between nodes. Now, in order to discover the DSTAP, is necessary to explain the two probabilistic association rules:

R_1 : "*low support & high confidence.*"

R_2 : "*high support & low confidence.*"

The rules will be applied to each relation of the form $P(\mathbf{X}[t]|Pa(\mathbf{X}[t]))$. Note that, in the case of initial graph B_0 the parents $Pa(\mathbf{X}[t])$ are in the same timestamp $t = 0$, and in the case of transition graph $B_{\rightarrow}$ the parents $Pa(\mathbf{X}[t])$ could be in the same timestamp t or the previous $t - 1$.

Mimicking the research work in [13] and providing a careful dynamical extension. The relationships "Parent-Children" described in $P(\mathbf{X}[t]|Pa(\mathbf{X}[t]))$, are called "*relational subspaces.*" These subspaces will provide the advantage of discovering meaningful temporal anomalies in a subspace level, through which reasons for anomalous temporal nature can also be contextualized and explained.

To explain the concept of relational subspaces, consider Figure 4–1, in which the unrolled dynamic Bayesian network model on timestamps $t-1, t$, and $t+1$ is shown. Let us fixed the timestamp t; in this, there exist three relational subspaces, i.e., $(X_1[t]|X_1[t-1])$, $(X_2[t]|X_1[t], X_1[t-1], X_2[t-1])$, and $(X_3[t]|X_2[t], X_2[t-1])$; we can rewrite this subspaces in a directed relational form, as: $\Big(X_1[t-1] \rightarrow X_1[t]\Big)$, $\Big(X_1[t], X_1[t-1], X_2[t-1] \rightarrow X_2[t]\Big)$, and $\Big(X_2[t], X_2[t-1] \rightarrow X_3[t]\Big)$ respectively.

In advance, we will refer the l-th relational subspaces as an "RS_l"; now it is relevant to mention the following notes:

- The rules R_1 and R_2 will be applied to each relational subspace RS_l, to discover low possibles patterns situated in each subspace.

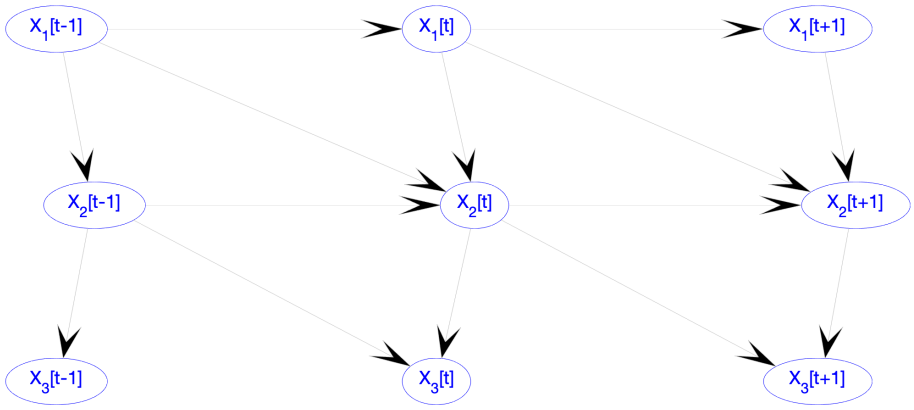

Figure 4–1: Unrolled dynamic Bayesian network on three timestamps $t-1, t$, and $t+1$, describing relational subspaces $(\mathbf{X}[t]|Pa(\mathbf{X}[t]))$.

- In the transition graph $B_{\rightarrow}$, a parent node $Pa(\mathbf{X}[t])$ in one relational subspace RS_l, could be as a child node in other relational subspace, in the same timestamp t or the previous $t-1$, e.g., in Figure 4–1, $X_2[t]$ is a parent node in $RS_{l_1} := \Big(X_2[t], X_2[t-1] \rightarrow X_3[t]\Big)$ but, $X_2[t]$ is a child node in $RS_{l_2} := \Big(X_1[t], X_1[t-1], X_2[t-1] \rightarrow X_2[t]\Big)$ in the timestamp t.
- In an arbitrary relational subspace RS_l in timestamp t, there could exist parents in the same timestamp t, or the previous $t-1$ for just one child.

Now, in order to discover the "*Domain Specific Temporal Anomalous Pattern*" (DSTAP) along of each timestamp t, let us define the rules as follows:

R_1 : In every relational subspace RS_l, choose the configurations in which the parents of a child node have "*low support*" & the child node has "*high confidence.*"

R_2 : In every relational subspace RS_l, choose the configurations in which the parents of a child node have "*high support*" & the child node has "*low confidence.*"

The rules are based on the concepts of *support* and *confidence*, both described in section 3.3. In the temporal context, we can extend both concepts as:

Definition: $support(X_i[t]) = P(X_i[t] = j)$

Definition: $confidence(X_i[t]) = P\big(X_i[t] = j | Pa(X_i[t]) = k\big)$

Where $i = 1, \ldots, n$; $j = 1, \ldots, r_i$; $k = 1, \ldots, q_i$; and $t = 0, \ldots, T$

Since the dynamic Bayesian network model $(B_0, B_{\rightarrow})$ was already learned, both *support* and *confidence* are available for every node in the model.

Now we can describe the *support* and *confidence* in each relational subspace RS_l, the *support* for the parent nodes, and the *confidence* for each child node given its parents in each RS_l. Thus, the specific concepts of *support* and *confidence* in each RS_l are shown in equation 4.1 and 4.2.

$$support\Big(X_i[t]\Big)_{X_i[t] \in RS_l} = P\Big(X_i[t] = j\Big)_{X_i[t] \in RS_l} \tag{4.1}$$

$$confidence\Big(X_i[t]\Big)_{X_i[t] \in RS_l} = P\Big(X_i[t] = j \Big| Pa\big(X_i[t]\big) = k\Big)_{Pa(X_i[t]), X_i[t] \in RS_l} \tag{4.2}$$

Both rules will provide patterns with no significative evidence to accept them as a typical pattern in specific domain knowledge. Those mined patterns will be patters whose "cause" are low probable, but with a high impact in the "effect" inside each RS_l, according to the rule R_1. Patterns whose "cause" are highly probable, but with low impact in the "effect" inside each RS_l, according to the rule R2. In both scenarios described above, there exists a natural conflict on the flow information into the modeling process.

Table 4–1: Abbreviations of Support and Confidence.

Low Support := *minsupp*	**Low Confidence** := *minconf*
High Support := *maxsupp*	**High Confidence** := *maxconf*

Note that both rules are based on high or maximum and low or minimum; thus, the four combinations are shortened and presented in Table 4–1. Since the dynamic Bayesian network was learned, every parameter of a node is available; thus, the combination *minsupp* is specified by the network. Defined as the j-th configuration (category) of each node $X_i[t]$ in which, its probability $P(X_i[t])$ is the lowest of all configurations r_i. Similarly, *maxsupp* is defined as the j-th configuration of each node $X_i[t]$ in which, its probability $P(X_i[t])$ is the greatest of all configurations r_i. Formal definitions of *minsupp* and *maxsupp* inside of each RS_l are provided in equations 4.3 and 4.4.

$$minsupp\Big(X_i[t]\Big)_{X_i[t]\in RS_l} = \arg\min_j P\Big(X_i[t]=j\Big)_{X_i[t]\in RS_l} \tag{4.3}$$

$$maxsupp\Big(X_i[t]\Big)_{X_i[t]\in RS_l} = \arg\max_j P\Big(X_i[t]=j\Big)_{X_i[t]\in RS_l} \tag{4.4}$$

In order to discover patterns with non-significative evidence to accept them as a typical pattern, the two remaining combinations, *minconf*, and *maxconf* must be defined by the user as a "*thresholds.*"

Keeping in mind that, the parents of a node is a set of nodes, and with these four combinations set up, the formal definition of the two rules are rewritten in equations 4.5 and 4.6.

$$\mathbf{R_1} := \Bigg[\Big(P\big(X_{pa}=k\big)=minsupp\Big)\wedge\Big(P\big(X_i[t]=j\big|X_{pa}\big)>maxconf\Big)\Bigg]_{\forall X_{pa}\in Pa(X_i[t])\in RS_l} \tag{4.5}$$

$$\mathbf{R_2} := \Bigg[\Big(P\big(X_{pa}=k\big)=maxsupp\Big)\wedge\Big(P\big(X_i[t]=j\big|X_{pa}\big)<minconf\Big)\Bigg]_{\forall X_{pa}\in Pa(X_i[t])\in RS_l} \tag{4.6}$$

Finally, applying both rules $\mathbf{R_1}$ and $\mathbf{R_2}$ in each relational subspace RS_l, we can discover each "*Domain Specific Temporal Anomalous Pattern*" (DSTAP) in a

relational form:

$$\big(X_{pa} = k\big) \rightarrow \big(X_i[t] = j\big) \tag{4.7}$$

Where $X_{pa} \in Pa(X_i[t])$ represent parent nodes, k and j represent specific configurations (categories) taken by parent and child nodes respectively, satisfying both rules 4.5 and 4.6, in the timestamp t.

To describe the process of the applicability of both rules, we show a small hypothetical example. Consider the structure of a dynamic Bayesian network model $(B_0, B_{\rightarrow})$ described in Figure 4–2, in timestamps $t = 0$ and $t = 1$.

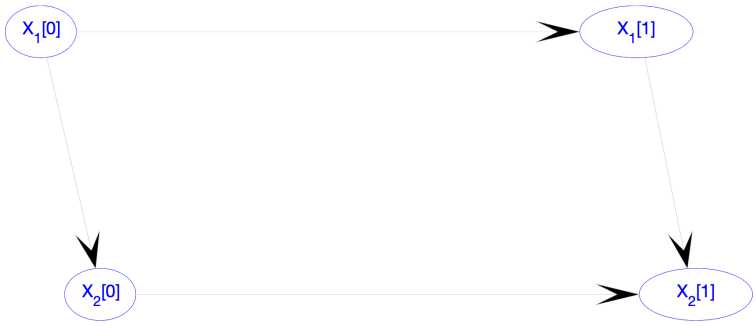

Figure 4–2: Hypothetical dynamic Bayesian network model $(B_0, B_{\rightarrow})$, on time $t = 0, 1$.

The network $(B_0, B_{\rightarrow})$, describes two temporal markovian sequences in the two first timestamps. There are four nodes $X_1[0], X_2[0], X_1[1]$, and $X_2[1]$. Assuming that each sequence is discrete with two states: true $= T$ and false $= F$; thus, the conditional probability table (CPT) of each node is described in Tables 4–2, 4–3, 4–4, 4–5.

Table 4–2: Hypothetical CPT of $X_1[0]$.

$X_1[0]$	
F	T
0.90	0.10

Table 4–3: Hypothetical CPT of $X_2[0] \big| X_1[0]$.

$X_2[0]$		
$X_1[0]$	F	T
F	0.05	0.95
T	0.15	0.85

Table 4–4: Hypothetical CPT of $X_1[1]\big|X_1[0]$.

$X_1[1]$		
$X_1[0]$	F	T
F	0.92	0.08
T	0.22	0.78

Table 4–5: Hypothetical CPT of $X_2[1]\big|X_2[0], X_1[1]$.

$X_2[1]$			
$X_2[0]$	$X_1[1]$	F	T
F	F	0.30	0.70
T	F	0.50	0.50
F	T	0.97	0.03
T	T	0.60	0.40

Note that the table distribution for parent node $X_1[0]$ is unconditional, instead of child nodes are conditionals. An additional remark is that, child nodes $X_2[0]$ and $X_1[1]$ from the parent node $X_1[0]$ in timestamp $t = 0$, will be parent nodes in timestamp $t = 1$.

In the network $(B_0, B_{\rightarrow})$ from Figure 4–2, there exist three relational subspaces:

$$RS_1 = \Big(X_1[0] \rightarrow X_2[0]\Big)$$
$$RS_2 = \Big(X_1[0] \rightarrow X_1[1]\Big)$$
$$RS_3 = \Big(X_1[1], X_2[0] \rightarrow X_2[1]\Big)$$

Rules $\mathbf{R_1}$ and $\mathbf{R_2}$ must be applied to the three relational subspaces. Setting up, the user parameters $minconf = 10\%$ and $maxconf = 80\%$

- On RS_1, applying 4.3 and 4.4, then, the network parameters $minsupp(X_1[0]) = T$ and $maxsupp(X_1[0]) = F$ respectively. The form of the rules $\mathbf{R_1}$ and $\mathbf{R_2}$ described in 4.5 and 4.6 will be:

$$\mathbf{R_1} = \left[\Big(P\big(X_1[0] = T\big) = 10\%\Big) \wedge \Big(P\big(X_2[0] = j\big|X_1[0] = T\big) > 80\%\Big)\right]$$
$$\mathbf{R_2} = \left[\Big(P\big(X_1[0] = F\big) = 90\%\Big) \wedge \Big(P\big(X_2[0] = j\big|X_1[0] = F\big) < 10\%\Big)\right]$$

Finally, applying $\mathbf{R_1}$ and $\mathbf{R_2}$ on RS_1, we obtain two "*Domain Specific Temporal Anomalous Pattern*" (DSTAP) of form 4.7:

$\big(X_1[0] = T\big) \rightarrow \big(X_2[0] = T\big)$ example of $\mathbf{R_1}$.

$\big(X_1[0] = F\big) \rightarrow \big(X_2[0] = F\big)$ example of $\mathbf{R_2}$.

- On RS_2, applying both rules $\mathbf{R_1}$ and $\mathbf{R_2}$, we obtain one DSTAP:
  $\big(X_1[0] = F\big) \rightarrow \big(X_1[1] = T\big)$ example of $\mathbf{R_2}$.

- On RS_3, both parent nodes $X_1[1]$ and $X_2[0]$, was previously child nodes; now applying 4.3 and 4.4, we obtain: $minsupp(X_2[0]) = F$, $minsupp(X_1[1]) = T$, $maxsupp(X_2[0]) = T$ and $maxsupp(X_1[1]) = F$. Then, applying both rules $\mathbf{R_1}$ and $\mathbf{R_2}$, we obtain one DSTAP:
  $\Big[\big(X_2[0] = F\big), \big(X_1[1] = T\big)\Big] \rightarrow \big(X_2[1] = F\big)$ example of $\mathbf{R_1}$.

In this way, we discovered in total four DSTAPs on timestamp $t = 0, 1$ in the dynamic Bayesian network model described in Figure 4–2. This process of discovering a DSTAP will be along the time observation $t = 0, \ldots T$. As a final remark, any instance belonging to the test data in the process of discovering meaningful temporal anomalies will be check if the instance fulfill any DSTAP e.i, a simple **selection** of each test instance that satisfy any of the discovered DSTAPs will be considered as an interesting anomaly, e.g., in the hypothetical example described in Figure 4–2, in $t = 0$, the procedure will be:

SELECT (a test instance) **IF** (the test instance satisfy):

$\Big(X_1[0] = T$ **AND** $X_2[0] = T\Big)$ **OR** $\Big(X_1[0] = F$ **AND** $X_2[0] = F\Big)$

In the same way for others timestamps $t = 1$, or a couple of timestamps $t, t + 1$.

In summary, our main contribution to the methodology DSTAP will discover interesting temporal anomalies in datasets. However, there could exist the main drawback that is the false positive rate, because, in any arbitrary network, we can

get a large number of DSTAP?s (|DSTAP|). Sensitivity analysis in Bayesian networks is performed to avoid this issue. Technically, the sensitivity analysis is the understanding of the relationship of parameters from the network with conclusions drawn from the network [52]. Thus, the proposal is to rank every DSTAP according to how interesting they are if the condition $|\text{DSTAP}| > 2n$ is fulfilled on each couple of timestamp t and $t+1$, according to the model $(B_0, B_{\rightarrow})$. To score every DSTAP of the form $\big(X_{pa} = k\big) \rightarrow \big(X_i[t] = j\big)$, based on a sensitivity measure, the instances in the nodes on the left side of the DSTAP are entered in the model and, the sensitivity measure is computed for the right side node (just a single node). Thus we get a score for every DSTAP.

4.3 Algorithm: Domain Specific Temporal Anomalous Patterns

The procedure for discovering the "Domain Specific Temporal Anomalous Patterns" (DSTAP) is presented in Algorithm 5. the input elements are the learned dynamic Bayesian network model $(B_0, B_{\rightarrow})$. The *minconf* and *maxconf* parameters, which are the minimum and maximum confidence for all child nodes, respectively. The number of temporal nodes n. The threshold parameter δ to consider the most relevant DSTAPs.

The Algorithm 5 shows a nested loop, thus the time complexity for this part is governed by T the length of the temporal sequences, and l the number of relational subspaces RS in the model $(B_0, B_{\rightarrow})$. Using both rules, in step 4, is constant computational time. The sensitivity analysis in Bayesian networks is posed as an NP-complete problem in the worst case. However, in step 5, sensitivity is using very sparsely in each DSTAP; thus, it took at most l times, then the order complexity is $O(Tl^2)$. Note that the if statements are neglected since the nested loop bounds it.

Algorithm 5 DSTAP.

Input: $DBN = (B_0, B_{\rightarrow})$, *minconf*, *maxconf*, n, δ, test data.
Output: DSTAP, temporal anomalies.

```
for all pair (t, t+1) from 0 to T do
    Find minsupp and maxsupp for parent nodes in (B_0, B_→) by 4.3 and 4.4
    for all Relational subspace RS_l in (B_0, B_→) do
        Use R_1 and R_2 by 4.5 and 4.6 to find all DSTAP of the form 4.7
        Calculate the sensitivity measure for discovered DSTAP according [52].
    end for
end for
if |DSTAP| > 2n then
    Sort all DSTAPs {according its sensitivity measure.}
    print Top (δ × |DSTAP|) {low scored DSTAP}
else
    print all discovered DSTAP
end if
if a test instance satisfy any DSTAP then
    print test instance as interesting temporal anomaly
end if
```

4.4 Efficiency of the DSTAP Methodology

The process of discovering interesting temporal anomalies from the dataset is based on the theory of dynamic Bayesian network models and the DSTAP methodology.

The learning process in the dynamic Bayesian network models integrates the relationships among stochastic processes of a specific domain, and the belief of and event in probabilistic terms, as we studied in Section 3.2.

The DSTAP methodology depends on the rules $\mathbf{R_1}$ that uses the parameter *maxconf*, and $\mathbf{R_2}$ that uses the parameter *minconf*; both parameters are independent of each other, and a small variation in those parameters does not affect the mining process of DSTAP significantly. The efficiency of both rules $\mathbf{R_1}$ and $\mathbf{R_2}$ is based on the joint probability distribution (JPD). JDP is the product of priors and conditionals probabilities. The scenarios where there is a conflict among the evidence and the conditional probability of the event, are scenarios indicating potential outlying situations and are provided for the theory of dynamic Bayesian network and

our prior belief of the domain-specific situation. In [53], the scenarios where there is a conflict ($\mathbf{R_1}$ and $\mathbf{R_2}$) are called "mere" and "suspicious" coincidence, respectively. Defining a coincidence as "An event that provides support for an alternative to a currently favored causal theory, but not necessarily enough support to accept that alternative in light of its low prior probability." Thus, both rules $\mathbf{R_1}$ and $\mathbf{R_2}$ helps to discover "interestingly rare temporal patterns" instead of mining "irrelevant temporal patterns or temporal noise." Besides, the DSTAP methodology provides contextual information (in the form of relational subspaces) or explanation from the discovered temporal anomalies, a relevant property that, none method of temporal outlier detection reports, as far as we have known.

The related research work with this thesis, described in section 2.4, has two main drawbacks. First, most of these works do not perform the dynamic structure learning from datasets; instead, they assume a fixed dynamical network structure extracted subjectively for an expert. Second, all these works discover anomalies using the JPD from a DBN model. Making those works computationally intensive because to compute the JPD requires to perform dynamic inference, which is an NP-Complete problem. Instead, our DSTAP methodology does not require to compute the JPD; it only searches for conflict scenarios, described on rules $\mathbf{R_1}$ and $\mathbf{R_2}$. Thus, the DSTAP methodology is declared efficient.

Chapter 5 EXPERIMENTAL STUDY

This chapter reports the experimentation process performed in order to discover domain specific temporal anomalous patterns. With those patterns, we reported interesting temporal outliers and explaining them in temporal datasets (discrete sequences or time series) using coupled dynamic Bayesian networks and probabilistic association rules.

5.1 General Experimental Protocol

The process of interesting temporal anomaly detection in this thesis required two subprocesses. First, learning a dynamic Bayesian network from a dataset. Second, applying the DSTAP algorithm. The experimentation process for the algorithms described before was implemented by using and extending the "*Bayes Net Toolbox for MATLAB*" [54]. First, to reveal a causal probabilistic relationship between the discrete sequences or time series, technically, this represents learning the dynamic Bayesian network, and second, to discover the interesting temporal anomalies based on the DSTAP algorithm. Experiments were carried out on a dedicated PC with Intel Core i7 2.5 GHz, 64 bits architecture, four cores, 8 GB RAM-memory and under macOS Mojave. In general, the proposed DSTAP methodology runs on unsupervised mode; however, for evaluation purposes, we run some experiments in a supervised mode.

In a supervised scenario, the class labels (normal, anomaly) were assigned in the temporal dataset, where 80% of both classes datasets were used to train the dynamic Bayesian networks. The rest 20% of both classes dataset was for testing the

DSTAPs. In a static scenario, an anomaly detection method to reach high accuracy, must be employed to discover those test instances, which belong to a different class than the class used to train the model. Instead, in a dynamic scenario, model is trained with both classes; due to the intrinsic time dependency of the data.

In a unsupervised scenario, we set 80% of the dataset to train the dynamic Bayesian networks and discover the DSTAPs; the rest 20% for testing purposes was dedicated to discovering interesting temporal anomalies.

In temporal data, the training dataset represents the first 80%, ordered according to time. In the supervised mode, the main reason to experiment in a training and testing scenario is to stand out that *relational implication* is relevant in discovering interesting outliers. Instances belonging to different classes may encode different relational implications between the random variables only on intra timestamps but not on inter timestamps. For example, in the timestamp t, the relational implication $X_1[t] \rightarrow X_2[t]$ in a particular class may appear like $X_2[t] \rightarrow X_1[t]$ in another class with different probabilistic parameters. However, in the timestamps t and $t+1$, the relation implication $X_1[t] \rightarrow X_2[t+1]$ in a certain class is not possible to appear as $X_2[t+1] \rightarrow X_1[t]$ in another class, due to the way of learning the dynamic structure of the network (the ordered nature of time).

5.2 Toy Example

Consider the simplest case, a stochastic process composed of two discrete random sequences $X_1[t]$ and $X_2[t]$, each one with two categories (false $= F$, true $= T$). Suppose that both sequences are related according to the structure of a dynamic Bayesian network model described in Figure 5–1. Assuming that the process is stationary and Markovian, where the sequences $X_1[t]$ and $X_2[t]$, are related intra timestamps in the form $X_1[t] \rightarrow X_2[t]$ and are related inter timestamps in a first-order Markovian way into each sequence. The parameters of the network model are

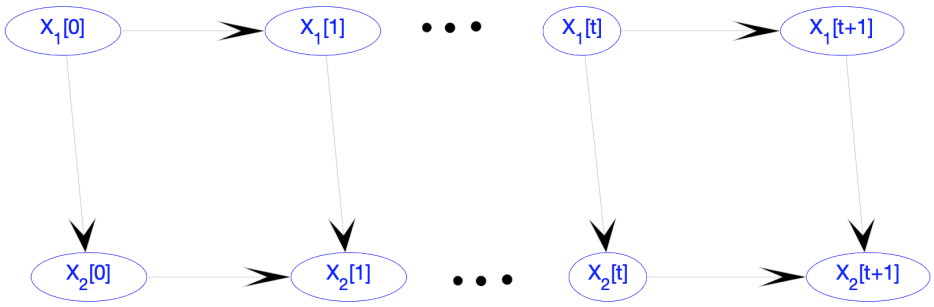

Figure 5–1: Toy example: unrolled dynamic Bayesian network model

defined for both networks: B_0 and $B_{\rightarrow}$ as:

- Parameters for B_0 for timestamp $t = 0$ are shown in Tables 5–1 and 5–2.

Table 5–1: Toy example CPT of $X_1[t]$.

$X_1[t]$	
F	T
0.90	0.10

Table 5–2: Toy example CPT of $X_2[t]\big|X_1[t]$.

$X_2[t]$		
$X_1[t]$	F	T
F	0.05	0.95
T	0.15	0.85

- Parameters for $B_{\rightarrow}$ for timestamps $t = 1, 2, ..., T$ are shown in Tables 5–3 and 5–4.

Table 5–3: Toy example CPT $X_1[t]\big|X_1[t-1]$.

$X_1[t]$		
$X_1[t-1]$	F	T
F	0.92	0.08
T	0.22	0.78

Table 5–4: Toy example $X_2[t]|X_2[t-1], X_1[t]$.

$X_2[t]$			
$X_2[t-1]$	$X_1[t]$	F	T
F	F	0.30	0.70
T	F	0.50	0.50
F	T	0.97	0.03
T	T	0.60	0.40

Let us call this model as $DBN_1 = (B_0, B_{\rightarrow})$. Once the dynamic Bayesian network model is set up, we present a simulation study by generating some training dataset from the model DBN_1, with different configurations, e.i., T that represents the number of timestamps, and $nseqs$ that represents the number of sequences or repetitions. When the dataset is in a traditional format like a matrix with each row representing a discrete random sequence $X_i[t]$, and each column representing a timestamp t, then $nseqs = 1$. With those configurations, we can obtain a simulated temporal dataset to learn the structure and parameters of the dynamic Bayesian network model; after the network was learned, the DSTAPs and the interesting temporal outliers were discovered.

- We have been generating a training temporal dataset from the model DBN_1, with $T = 100$, $nseqs = 30$, and a fixed seed for repeatability purposes. With this synthetic data, first, we will reproduce the structure of the DBN_1 model using Algorithm 1. Second, learning the parameters with the MLE method, described in subsection 3.2.3. Third, discovering the DSTAPs using Algorithm 5. Finally, we will discover interesting temporal anomalies in unsupervised mode with a simulated testing dataset from the DBN_1 model.

  1. Learning the structure of the model DBN_1 from the synthetic temporal data: Assuming that the Intra timestamps relationship is the same in Bayesian networks B_0 and $B_{\rightarrow}$. The learned intra timestamps relationship adjacency matrix was:

  $$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

  Representing the directional relationship $X_1[t] \rightarrow X_2[t]$ for each $t = 0, 1, \ldots, T$. The learned inter timestamps relationship transition matrix was:

  $$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Representing the relationships $X_1[t-1] \rightarrow X_1[t]$ and $X_2[t-1] \rightarrow X_2[t]$ for $t = 1, \ldots, T$.

2. Learning the parameters of the model DBN_1 from synthetic temporal data:
   - Parameters learned for B_0 for timestamp $t = 0$, are shown in the Tables 5–5 and 5–6.

Table 5–5: Learned CPT of $X_1[t]$.

$X_1[t]$	
F	T
0.8667	0.1333

Table 5–6: Learned CPT of $X_2[t] \mid X_1[t]$.

$X_2[t]$		
$X_1[t]$	F	T
F	0.00	1.00
T	0.25	0.75

   - Parameters learned for $B_{\rightarrow}$ for timestamps $t = 1, 2, ..., T$, are shown in Tables 5–7 and 5–8.

Table 5–7: Learned CPT of $X_1[t] \mid X_1[t-1]$

$X_1[t]$		
$X_1[t-1]$	F	T
F	0.9184	0.0816
T	0.2030	0.7970

Table 5–8: Learned CPT of $X_2[t] \mid X_2[t-1], X_1[t]$.

$X_2[t]$			
$X_2[t-1]$	$X_1[t]$	F	T
F	F	0.3019	0.6981
T	F	0.4967	0.5033
F	T	0.9522	0.0478
T	T	0.5918	0.4082

3. We set the parameters from the Algorithm 5 as: $minconf$=10%, $maxconf$=90% and $\delta = 50\%$. After the Algorithm 5 was executed in the DBN_1 model: There are three relational subspaces, because the stochastic process is stationary for $t = 0, 1, \ldots, T-1$.

$$RS_1 = \Big(X_1[t] \rightarrow X_2[t]\Big)$$
$$RS_2 = \Big(X_1[t] \rightarrow X_1[t+1]\Big)$$
$$RS_3 = \Big(X_1[t+1], X_2[t] \rightarrow X_2[t+1]\Big)$$

There are three DSTAPs.

(a) $\big(X_1[t] = F\big) \rightarrow \big(X_2[t] = F\big)$ example of $\mathbf{R_2}$ on RS_1.

(b) $\big(X_1[t] = F\big) \rightarrow \big(X_1[t+1] = T\big)$ example of $\mathbf{R_2}$ on RS_2.

(c) $\Big[\big(X_2[t] = F\big), \big(X_1[t+1] = T\big)\Big] \rightarrow \big(X_2[t+1] = F\big)$ example of $\mathbf{R_1}$ on RS_3.

In this case, since the number of DSTAPs is smaller than $2(n) = 4$, there is no need to calculate the sensitivity measure.

4. Testing dataset to discover interesting temporal anomaly instances in an unsupervised mode. Assuming that the test dataset was given with $T = 20$ and $nseqs = 1$, for the specific temporal domain on which the model DBN_1 was previously trained. For this unsupervised task, we check if a test instance carries any of the three discovered DSTAPs. Figure 5–2 shows a simulated

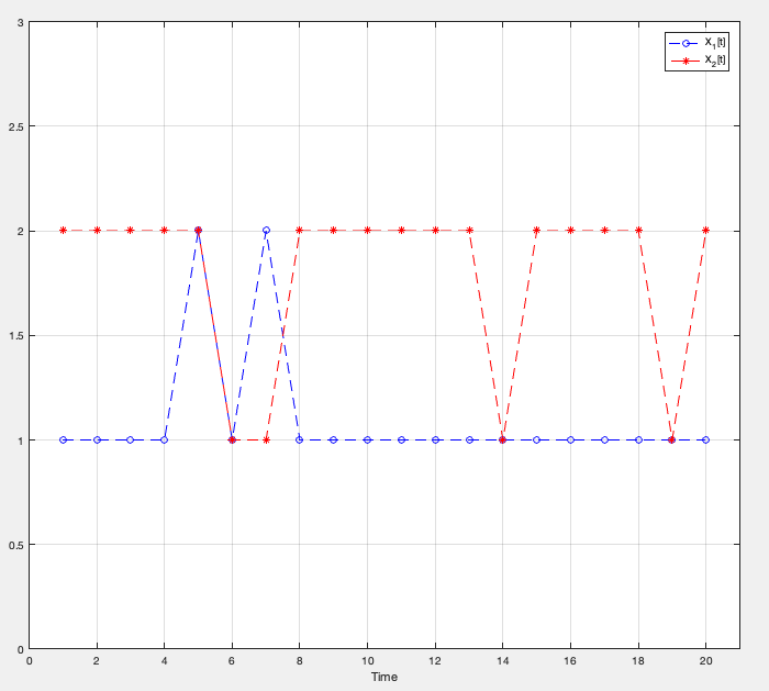

Figure 5–2: Test dataset from the model DBN_1.

testing temporal dataset from the DBN_1 model, the blue discrete sequence corresponds to the evolution of the stochastic process $X_1[t]$, and the red one corresponds to $X_2[t]$, both features are discrete, each with two categories false: $F = 1$, and true: $T = 2$.

DSTAPs are applied to test datasets in order to check if any test instances fulfill any of the DSTAPs.

Table 5–9: Instantaneous Outliers Associated to $DSTAP_1$.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Outliers	0	0	0	0	0	6	0	0	0	0	0	0	0	14	0	0	0	0	19	0

Table 5–9, shows the static (instantaneous) interesting outliers, related to the $DSTAP_1 := \big(X_1[t] = F\big) \rightarrow \big(X_2[t] = F\big)$. According to this, test instances with values $X_1[t] = F$ and $X_2[t] = F$ for $t = 1, 2, \ldots, 20$, are interesting static outliers, since they occur in a time instant t. Thus, three static outliers were discovered in time $t = 6$, $t = 14$, and $t = 19$, associated with the random process $X_2[t]$. The explainability of why they happened, is related to the relational subspace $RS_1 := \Big(X_1[t] \rightarrow X_2[t]\Big)$. Because the parent node $X_1[t]$ has a high prior probability of taking the false value F, child node $X_2[t]$ has a low conditional probability of taking the false value F. Thus, this basic scenario represent a conflict on the normal behavior of the domain knowledge captured by the DBN_1 model.

Table 5–10: Temporal Outliers Associated to $DSTAP_2$.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Outliers	0	0	0	0	5	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5–10, shows the interesting temporal outliers, related with the $DSTAP_2 := \big(X_1[t] = F\big) \rightarrow \big(X_1[t+1] = T\big)$, test instances with values $X_1[t] = F$ and

$X_1[t+1] = T$ for $t = 1, 2, \ldots, 20$, are the outliers corresponding to the random process $X_1[t]$. Two reported outliers in time $t = 5$, and $t = 7$. Note that, the child node $X_1[t+1]$, has a parent node $X_1[t]$; thus, the contextualization of why the interesting temporal outliers where mined, is because $X_1[t]$ has a high prior probability of taking the value false F and conversely the child node $X_1[t+1]$ has a low conditional probability of taking the value true T.

Table 5–11: Temporal Outliers Associated to $DSTAP_3$.

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Outliers	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5–11, shows the interesting temporal outliers, related with the $DSTAP_3 := \Big[\big(X_2[t] = F\big), \big(X_1[t+1] = T\big)\Big] \rightarrow \big(X_2[t+1] = F\big)$, test instances with values $X_2[t] = F$, $X_1[t+1] = T$, and $X_2[t+1] = F$ for $t = 1, 2, \ldots, 20$, are the outliers corresponding to the random process $X_2[t]$. There is one reported outlier in time $t = 7$. Note that, the child node $X_2[t+1]$, has two parent nodes $X_2[t]$ and $X_1[t+1]$. Thus, the interesting temporal outlier where mined since $X_2[t]$ and $X_1[t+1]$ both have low prior probability of taking the false values F and true values T, respectively. On the other hand, the child node $X_2[t+1]$ has a high conditional probability of take the false value F.

- An experimental sensitivity analysis of parameters *maxconf* and *minconf* was performed as a simulation study over the model DBN_1.

  Figure 5–3 shows the impact of the parameter *maxconf* varying from 0.7 to 0.99, over the number of discovered DSTAPs related to rule $\mathbf{R_1}$. When *maxconf* is permissive (between 0.70 and 0.75), the number of DSTAPs is three or two; hence, the amount of reported instantaneous or temporal outliers will probably be high; moreover, the method may fall in a high false positive. It will affect the precision

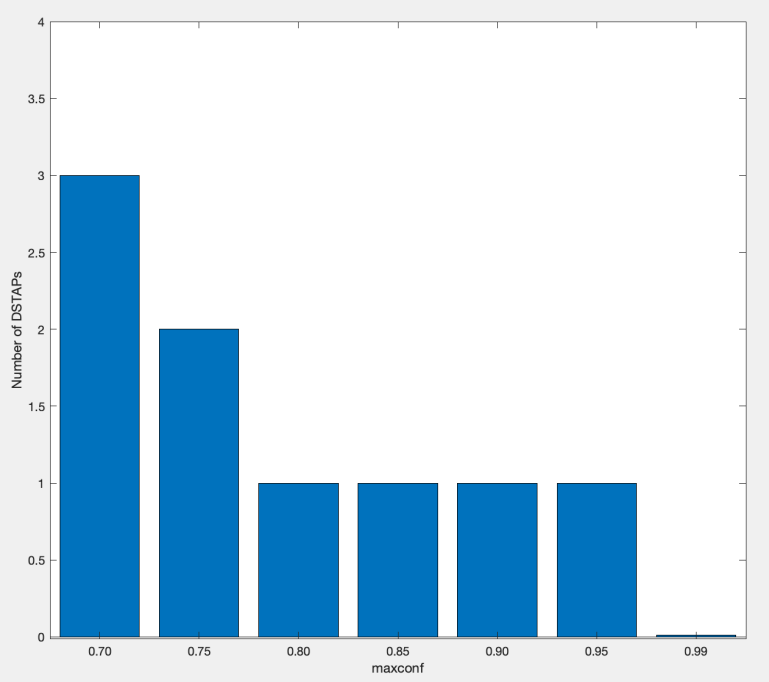

Figure 5–3: Effect of the user parameter *maxconf* on the number of discovered DSTAPs.

of the method in a supervised scenario. Instead if *maxconf* is more restrictive (between 0.80 and 0.99), the number of DSTAPs is one or zero, then the amount of reported instantaneous or temporal outliers will be probably low; moreover, the method may fall in a high false negative, and will affect the recall of the method in a supervised scenario. Finally, it is necessary to preserve a trade-off of the parameter *maxconf*; empirically, the range of values of *maxconf* is between 0.80 and 0.95.

Figure 5–4 shows the impact of the parameter *minconf* varying from 0 to 0.5, over the number of discovered DSTAPs related to rule $\mathbf{R_2}$. When *minconf* is permissive (between 0.10 and 0.50), the number of DSTAPs is two or three. Instead, if *minconf* is more restrictive (between 0 and 0.05), the number of DSTAPs is zero or one. As in the case of *maxconf*, the precision and recall will be affected while setting up a threshold; empirically, the range of values of *minconf* is between 0.05 and 0.10. Finally, it is important to remark that the DSTAP methodology depends

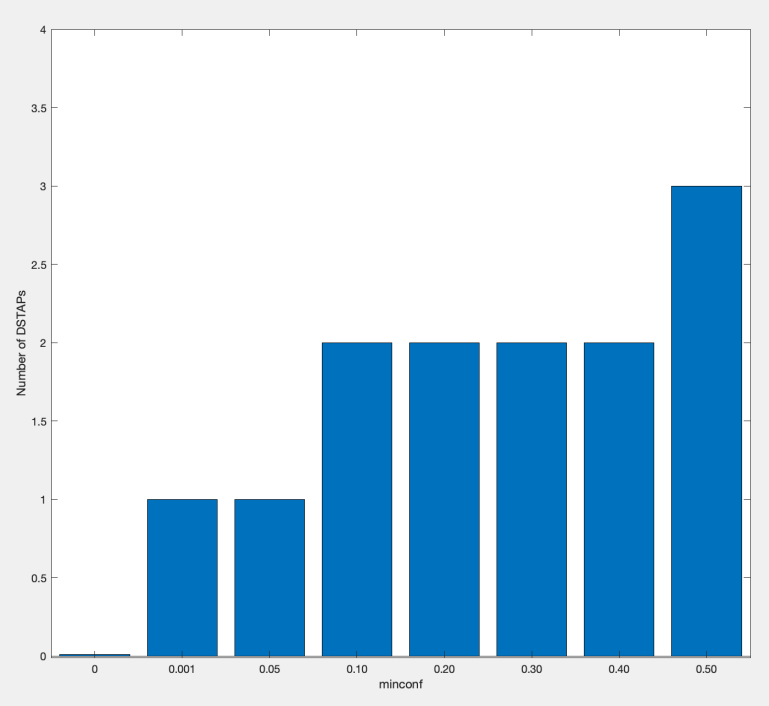

Figure 5–4: Effect of the user parameter *minconf* on the number of discovered DSTAPs.

highly on the previous learned dynamic Bayesian network. The structure and the parameters both are important since the topology affects the form of the *relational subspaces*, and the discovered DSTAPs. On the other hand, the parameters affect the number of the reported DSTAPs.

5.3 Synthetic Datasets

Now the DSTAP methodology on synthetic temporal datasets related to well-known DBN models can be employed. The results report interesting temporal outliers and their relational subspaces to explain the anomaly causes.

In [51], the author provides well-known dynamic Bayesian network models and their synthetic temporal datasets to benchmark structure learning algorithms. We have used four well-known dynamic Bayesian networks and their synthetic temporal datasets. First, we learned the structure and parameters of each network. Second, we discovered the DSTAPs for each model. Finally, we reported the interesting

temporal outliers provided by its probable anomaly causes.

Training temporal datasets were used to perform dynamic structure and parameters learning in four models, with $T = 500$, $nseqs = 50$. Table 5–12 shows the characteristics of the synthetic temporal datasets and a summary of the structure and parameters learned of each DBN from datasets. For example, the umbrella DBN model represents three discrete sequences (rain, umbrella, and height). For each variable, there are two categories. In alarm, there is some random process with two categories an others with 4. Moreover, the umbrella network presents five links learned, and 14 parameters learned. Finally, the time in seconds of the learning process is reported.

Table 5–12: Dynamic Structure and Parameters Learned from Synthetic Datasets.

Dynamic Nets.	**Sequences**	**Categories**	**Edges**	**Parameters**	**Time** (seg)
Umbrella	3	2	5	22	36
Cancer	5	2	12	58	185
Asia	8	2	21	100	450
Alarm	37	2-4	110	945	1.85×10^3

Figures 5–5, 5–6, and 5–7 show the structure learned of dynamic Bayesian networks from synthetic datasets, namely umbrella, cancer, and asia, respectively.

We set the parameters $minconf$=10%, $maxconf$=85% and $\delta = 50\%$. After Algorithm 5 was executed on the previous dynamic Bayesian networks learned from synthetic datasets, we summarize the information on the total number of DSTAPs discovered and the time taken in the discovery process. Table 5–13 shows the findings of the DSTAP methodology.

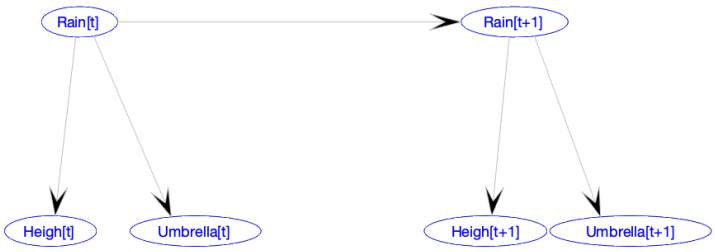

Figure 5–5: Structure learned Dynamic Bayesian network Umbrella.

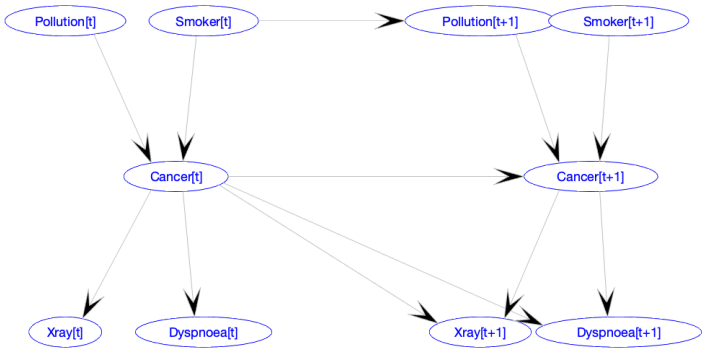

Figure 5–6: Structure learned Dynamic Bayesian network Cancer.

Table 5–13: Number of DSTAPs Discovered and Time on learned DBNs.

Dynamic Bayesian Networks	**Number of DSTAPs**	**Time** (seg)
Umbrella	5	5
Cancer	10	7
Asia	15	23
Alarm	65	244

The testing phase has been done to discover interesting temporal outlier instances in an unsupervised mode. Test data set is given with $T = 100$ and $nseqs = 1$. Table 5–14 shows the outliers, the DSTAPs, and the time slices where they appear

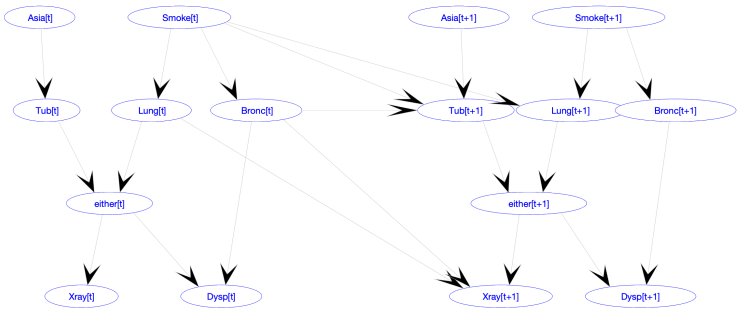

Figure 5–7: Structure learned Dynamic Bayesian network Asia.

for the umbrella DBN model.

Table 5–14: Relational Subspaces for outliers from Umbrella DBN model.

Outliers	**DSTAPs**	**Time slices**
Temporal	$(Rain[t] = T) \rightarrow (Height[t] = L)$	$t = 5:6, 87:89$
Global	$(Rain[t] = F) \rightarrow (Height[t] = H)$	$t = 24$
Points	$(Rain[t] = T) \rightarrow (Umbrella[t] = F)$	$t = 23$
and	$(Rain[t] = F) \rightarrow (Umbrella[t] = T)$	$t = 45:47$
Collectives	$(Rain[t] = T) \rightarrow (Rain[t+1] = F)$	$t = 5:7, 45:46$

The advantage of the DSTAP method is that it can both identify interesting temporal outliers and explain the reason for their unusual nature. According to Table 5–14, we can contextualize the findings as follow:

- In time slices $t = 5:6, 87:89$ the relational subspace:

  $(Rain[t] = T) \rightarrow (Height[t] = L)$

  Reports 5 interesting temporal outliers (global and collective points), on the random process Height with level low, moreover, the probable anomaly cause is the random process Rain with level true.

- In time slice $t = 24$ the relational subspace:

  $(Rain[t] = F) \rightarrow (Height[t] = H)$

Report 1 interesting temporal outliers (global point), on the random process Height with level high, moreover, the probable anomaly cause is the random process Rain with level false.

- In time slice $t = 23$ the relational subspace:

  $\big(Rain[t] = T\big) \rightarrow \big(Umbrella[t] = F\big)$

  Report 1 interesting temporal outliers (global point), on the random process Umbrella with level false, moreover, the probable anomaly cause is the random process Rain with level true.

- In time slices $t = 45 : 47$ the relational subspace:

  $\big(Rain[t] = F\big) \rightarrow \big(Umbrella[t] = T\big)$

  Report 3 interesting temporal outliers (global and collective points), on the random process Umbrella with level true, moreover, the probable anomaly cause is the random process Rain with level false.

- In time slice $t = 4 : 6, 44 : 45$ the relational subspace:

  $\big(Rain[t] = T\big) \rightarrow \big(Rain[t + 1] = F\big)$

  Report 5 interesting temporal outliers (global and collective points), on the random process Rain with level F at time slice $t+1$, moreover, the probable anomaly cause is the random process Rain with level true at the previous time slice t.

A total of 15 interesting temporal outliers was reported and explained according to their relational subspaces.

For space limitations, we present a summary of the discovered interesting temporal outliers on DBN models related to Cancer, Asia, and Alarm networks. Table 5–15 summarizes the findings of DSTAPs on Cancer, Asia, and Alarm networks.

Table 5–15: Summary of temporal outliers of DBNs Cancer, Asia and Alarm.

DBN	DSTAPs	Temporal Outliers	Time slices
Cancer	10	4	$t = 35 : 37, 45$
Asia	15	8	$t = 23, 37 : 39, 85 : 88$
Alarm	65	15	$t = 11, 45 : 49, 56 : 59, 96 : 100$

We have obtained four interesting temporal outliers from a sequence of 100 instances on DBN Cancer, eight interesting temporal outliers from a sequence of 100 instances on DBN Asia, and 15 interesting temporal outliers from a sequence of 100 instances on DBN Alarm. The contextualization of discovered outliers of the DBN Cancer, Asia, and Alarm must be done in the same manner as the previous DBN Umbrella.

5.4 Real Datasets

To assess the performance of the DSTAP methodology in a supervised mode in a specific, realistic scenario, we applied the method on multivariate time series. The temporal data comes from a secure water treatment (SWaT) system [55], which is a scaled-down version of a real-world industrial water treatment plant. Data collection was under two behavioral modes normal and attacked, the period was 11 days, and it was logged continuously once every second. Data recorded was obtained from the sensors and actuators. Sensors are devices that convert a physical parameter into an electronic output, i.e., an electronic value. In contrast, actuators are devices that convert a signal into a physical output, i.e., turning the pump off or on. In total, 946,722 samples comprising of 51 attributes were collected, 24 from attributes correspond to sensors, the remaining to actuators.

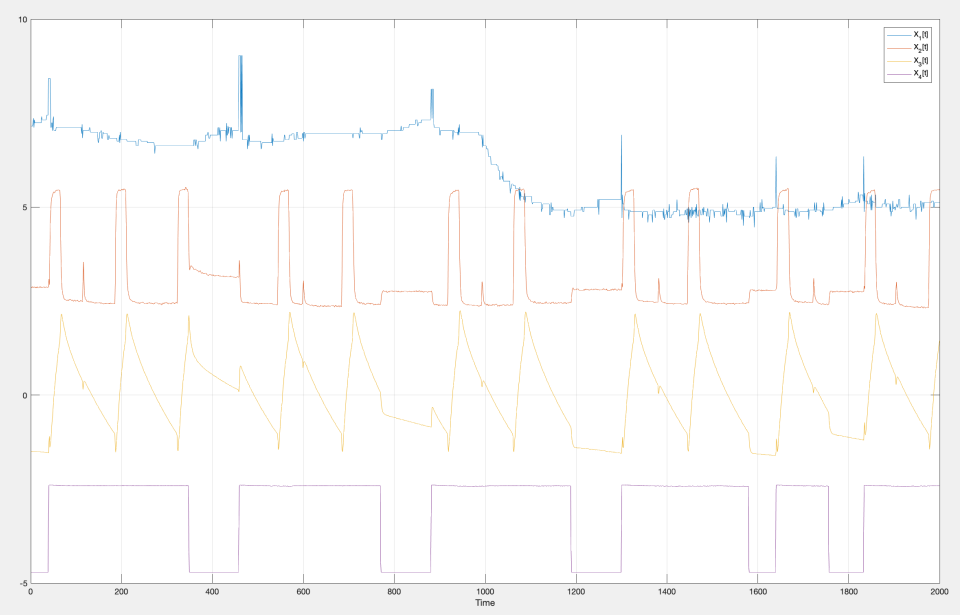

Figure 5–8: Time series from 4 sensors used from SWaT.

We compared our results with the research described in [56]. Unfortunately, the SWaT dataset[1] has labels not aligned with the outliers, because there is a delay between the attack and the actual disruption of the system. Instead, an attack-free version was found online, enabling us to perturb the data with the general types of temporal outliers. The attack-free dataset consists of 496800 instances and 24 time series from sensors in the SWaT. We used four sensors that measured the same system component, guaranteeing that they are sufficiently correlated. We extracted only a downsampled version of 2000 instances. The temporal data was previously standardized. For visual purposes, we separate the series, as shown in Figure 5–8.

- $X_1[t]$ =AIT-201: Conductivity analyzer; Measures NaCl level.
- $X_2[t]$ =AIT-202: pH analyzer; Measures HCl level.
- $X_3[t]$ =AIT-203: ORP analyzer; Measures NaOCl level.
- $X_4[t]$ =FIT-201: Flow Transmitter; Control dosing pumps.

[1] https://itrust.sutd.edu.sg

We perturbed the SWaT dataset by injecting temporal outliers. Injecting with either global points outliers, contextual points outliers, or contextual collective outliers. Sampling, random sequences of 5 data instances for global or contextual points outliers, and sampling, random sequences of 11 data instances for contextual collective outliers. We added 3σ to the feature values, and in some cases, flipped the sign of the data points or sequences, same as in [56]. Table 5–16 summarizes the outlier characteristics.

Table 5–16: Summary of the injected temporal outliers in the SWaT dataset.

Outlier type	**Outlying data points**
Global points	3×5
Contextual points	4×5
Contextual collective	5×11

In Figure 5–9 we show the perturbed 4 time series.

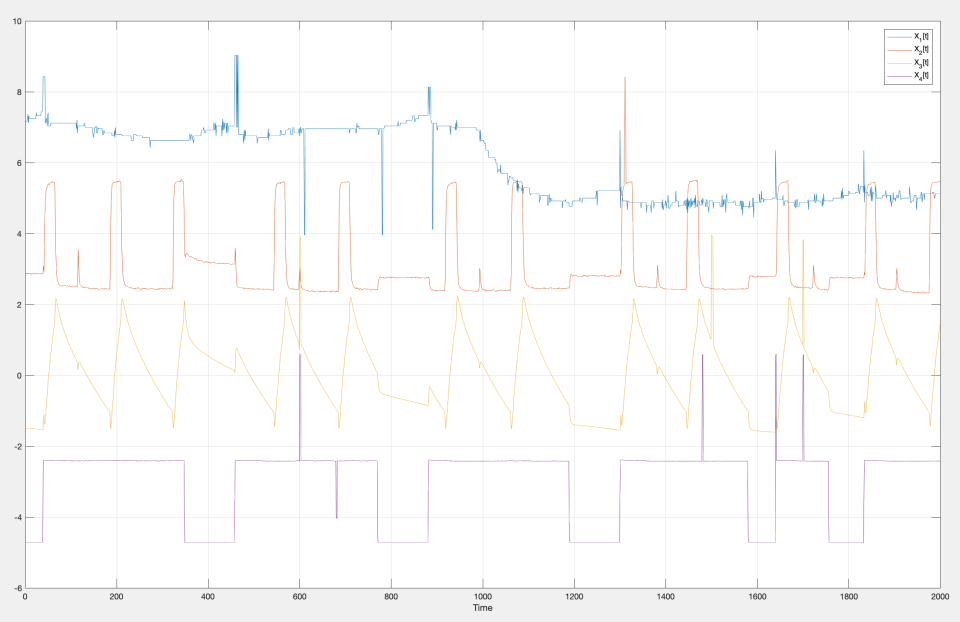

Figure 5–9: Perturbed time series from 4 sensors used from SWaT.

The perturbation process of temporal data was done to put the DSTAP methodology at work in a supervised mode. As discussed in Section 3.4, we discretized the data using binds with equal width. An extensive experimental evaluation suggests that the number of binds ranges between 4 and 9; we choose seven bins for $X_1[t], X_2[t]$ and $X_3[t]$, instead of for $X_4[t]$ we choose four bins (due to the nature of the random process). Table 5–17 shows the bins and edges of the discretization process

Table 5–17: Summary of the discretization in the SWaT dataset.

Series	**Bins and Edges**
$X_1[t]$	$1 = [258, 258.8), 2 = [258.8, 259.6), 3 = [259.6, 260.4), 4 = [260.4, 261.2)$ $5 = [261.2, 262), 6 = [262, 262.8), 7 = [262.8, 263.6)$
$X_2[t]$	$1 = [8.3, 8.42), 2 = [8.42, 8.54), 3 = [8.54, 8.66), 4 = [8.66, 8.78)$ $5 = [8.78, 8.9), 6 = [8.9, 9.02), 7 = [9.02, 9.14)$
$X_3[t]$	$1 = [410, 429), 2 = [429, 448), 3 = [448, 467), 4 = [467, 486)$ $5 = [486, 505), 6 = [505, 524), 7 = [524, 543)$
$X_4[t]$	$1 = [0, 0.9), 2 = [0.9, 1.8), 3 = [1.9, 2.7), 4 = [2.7, 3.6)$

We call the DBN model as DBN_{SWaT}. The learned dynamic structure is represented in Figure 5–10

Note that the structure of DBN_{SWaT}, suggests the time series are autocorrelated with a lag of order one and are cross-correlated within each timestamp. Moreover, The random process $X_4[t]$ =FIT-201: Flow transmitter, represents the parent node from the child $X_3[t]$ =AIT-203: ORP analyzer and $X_2[t]$ =AIT-202: pH analyzer; whereas, $X_3[t]$ =AIT-203: ORP analyzer is the parent of $X_1[t]$ =AIT-201: Conductivity analyzer. Those relations represent causal probabilistic dependency. The learned parameters of the model DBN_{SWaT} were performed. Due to space limitations, we present the two smallest CPTs in Tables 5–18 and 5–19, for $X_4[t]$ and

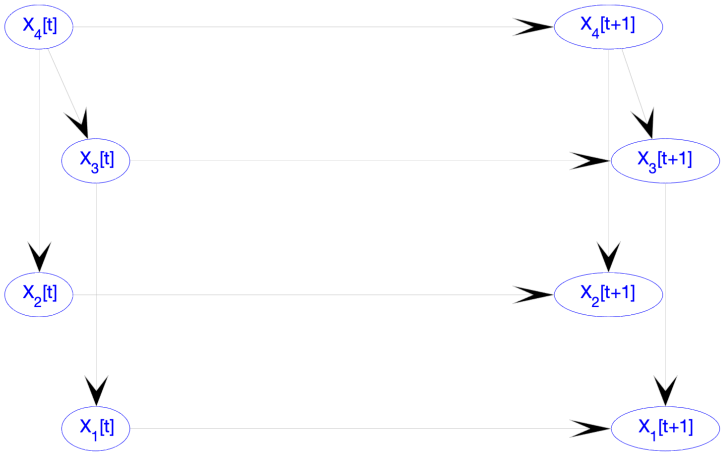

Figure 5–10: Dynamic structure learned of the DBN_{SWaT} model in timestamps t and $t+1$.

$X_4[t+1]\Big|X_4[t]$ respectively.

Table 5–18: Learned CPT of $X_4[t]$ from DBN_{SWaT}.

$X_4[t]$			
1	2	3	4
0.2520	0.0050	0.7390	0.0040

Table 5–19: Learned CPT of $X_4[t+1]\Big|X_4[t]$ from DBN_{SWaT}.

$X_4[t+1]$				
$X_4[t]$	1	2	3	4
1	0.9881	0.0079	0.0020	0.0020
2	0.3333	0.1111	0.5556	0.0000
3	0.0140	0.0270	0.9570	0.0020
4	0.0010	0.0010	0.1924	0.8056

An extensive experimental evaluation suggests that the threshold parameters are *minconf*=10%, *maxconf*=80%. After the Algorithm 5 was executed in the DBN_{SWaT} model:

There were seven relational subspaces

$$RS_1 = \Big(X_4[t] \rightarrow X_3[t]\Big) \qquad RS_5 = \Big(X_3[t], X_4[t+1] \rightarrow X_3[t+1]\Big)$$
$$RS_2 = \Big(X_4[t] \rightarrow X_2[t]\Big) \qquad RS_6 = \Big(X_2[t], X_4[t+1] \rightarrow X_2[t+1]\Big)$$
$$RS_3 = \Big(X_3[t] \rightarrow X_1[t]\Big) \qquad RS_7 = \Big(X_1[t], X_3[t+1] \rightarrow X_1[t+1]\Big)$$
$$RS_4 = \Big(X_4[t] \rightarrow X_4[t+1]\Big)$$

DSTAPs

1. $\big(X_4[t] = 4\big) \rightarrow \big(X_3[t] = 7\big)$ example of $\mathbf{R_1}$ on RS_1.
2. $\big(X_4[t] = 3\big) \rightarrow \big(X_3[t] = 7\big)$ example of $\mathbf{R_2}$ on RS_1.
3. $\big(X_4[t] = 3\big) \rightarrow \big(X_2[t] = 7\big)$ example of $\mathbf{R_2}$ on RS_2.
4. $\big(X_3[t] = 2\big) \rightarrow \big(X_1[t] = 1\big)$ example of $\mathbf{R_2}$ on RS_3.
5. $\big(X_4[t] = 4\big) \rightarrow \big(X_4[t+1] = 4\big)$ example of $\mathbf{R_1}$ on RS_4.
6. $\big(X_4[t] = 3\big) \rightarrow \big(X_4[t+1] = 4\big)$ example of $\mathbf{R_2}$ on RS_4.
7. $\Big[\big(X_3[t] = 7\big), \big(X_4[t+1] = 4\big)\Big] \rightarrow \big(X_3[t+1] = 7\big)$ example of $\mathbf{R_1}$ on RS_5.
8. $\Big[\big(X_3[t] = 2\big), \big(X_4[t+1] = 3\big)\Big] \rightarrow \big(X_3[t+1] = 7\big)$ example of $\mathbf{R_2}$ on RS_5.
9. $\Big[\big(X_1[t] = 1\big), \big(X_3[t+1] = 2\big)\Big] \rightarrow \big(X_1[t+1] = 1\big)$ example of $\mathbf{R_1}$ on RS_7.
10. $\Big[\big(X_1[t] = 5\big), \big(X_3[t+1] = 3\big)\Big] \rightarrow \big(X_1[t+1] = 1\big)$ example of $\mathbf{R_2}$ on RS_7.

Detecting Interesting Temporal Outliers:

We summarize the findings graphically. Note that the reported interesting temporal outliers by DSTAPs could represent novelties (points) or sequences, according to the process of injecting temporal outliers described above.

Figure 5–11, represents the interesting temporal outliers for the time series $X_1[t]$ Conductivity analyzer, discovered by the DSTAPs 4, 9, and 10.

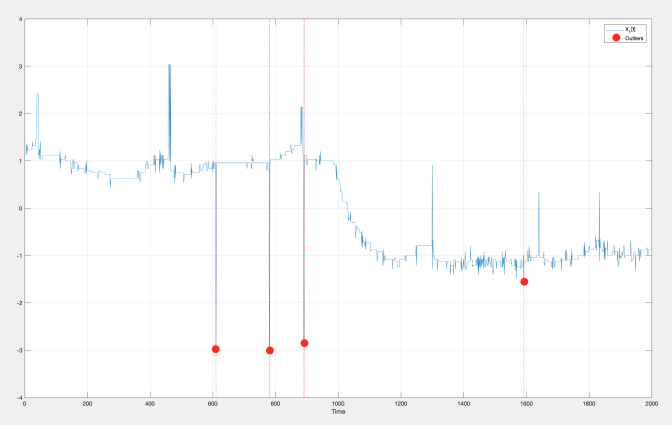

Figure 5–11: Interesting temporal outliers for time series $X_1[t]$: Conductivity analyzer.

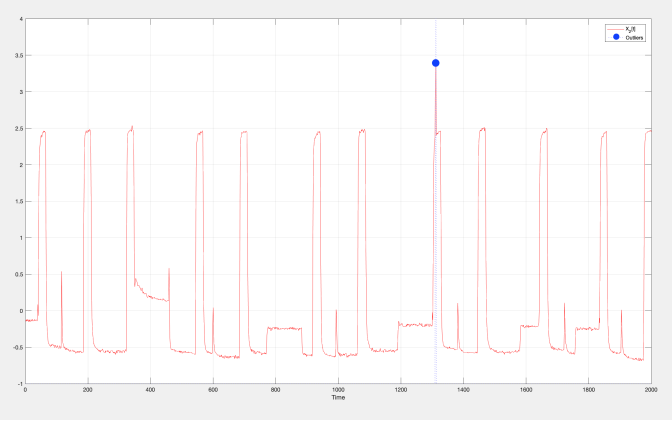

Figure 5–12: Interesting temporal outliers for time series $X_2[t]$: pH analyzer.

Figure 5–12, represents the interesting temporal outliers for the time series $X_2[t]$ pH analyzer, discovered by the DSTAP 3.

Figure 5–13, represents the interesting temporal outliers for the time series $X_3[t]$ ORP analyzer, discovered by the DSTAP 1, 2, 7, and 8.

Figure 5–14, represents the interesting temporal outliers for the time series $X_4[t]$ Flow transmitter, discovered by the DSTAP 5, and 6.

Contextualizing Interesting Temporal Outliers:

The advantage of the DSTAP method is that it can both identify interesting temporal outliers and explain the reason for their unusual nature.

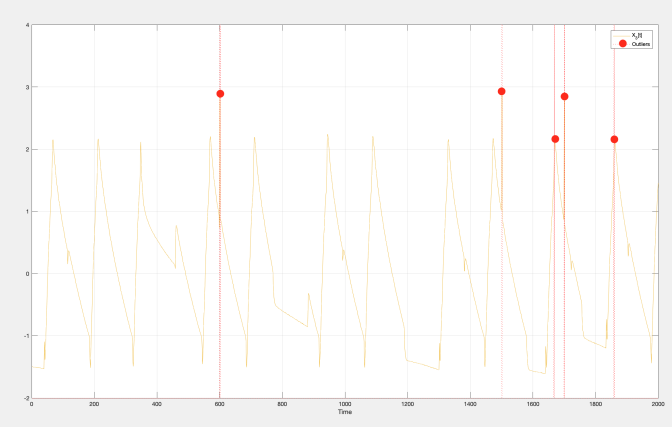

Figure 5–13: Interesting temporal outliers for time series $X_3[t]$: ORP analyzer.

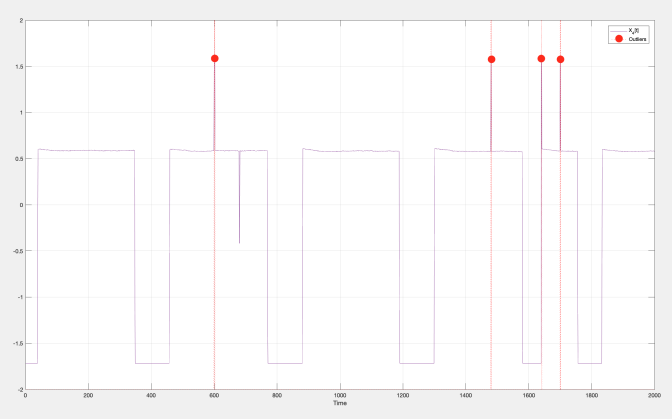

Figure 5–14: Interesting temporal outliers for time series $X_4[t]$: Flow transmitter.

The reported interesting temporal outliers from time series $X_1[t]$ Conductivity analyzer that measures the NaCl level is associated with the DSTAPs 4, 9, and 10. Table 5–20 shows the relational subspaces which were targeted by interesting temporal outliers for $X_1[t]$ presented in the SWaT dataset. For example:

- In time slices $t = 780 : 784, 890 : 894$ the relational subspace:

  $\big(ORP\ analyzer[t] = [429, 448)\big) \rightarrow \big(Conductivity\ analyzer[t] = [258, 258.8)\big)$

  Reports 10 interesting temporal outliers (global points), on the conductivity analyzer with levels of NaCl between $[258, 258.8)$, moreover, the probable anomaly cause is the ORP analyzer with leves of NaOCl between $[429, 448)$

- In time slices $t = 610 : 613$ the relational subspace:
  $\Big[\big(Conductivity\ analyzer[t] = [258, 258.8)\big), \big(ORP\ analyzer[t{+}1] = [429, 448)\big)\Big] \rightarrow \big(Conductivity\ analyzer[t+1] = [258, 258.8)\big)$
  Reports 4 interesting temporal outliers (global points), on the conductivity analyzer with levels of NaCl between $[258, 258.8)$, moreover, the probable anomaly causes are the ORP analyzer with leves of NaOCl between $[429, 448)$ in the same time slices, and the same conductivity analyzer with levels of NaCl in $[258, 258.8)$ in the previous time slice.
- In time slice $t = 609$ the relational subspace:
  $\Big[\big(Conductivity\ analyzer[t] = [261.2, 262)\big), \big(ORP\ analyzer[t{+}1] = [448, 467)\big)\Big] \rightarrow \big(Conductivity\ analyzer[t+1] = [258, 258.8)\big)$
  Reports 1 interesting temporal outlier (global point), on the conductivity analyzer with levels of NaCl between $[258, 258.8)$, moreover, the probable anomaly causes are the ORP analyzer with leves of NaOCl between $[448, 467)$ in the same time slices, and the same conductivity analyzer but with levels of NaCl between $[261.2, 262)$ in the previous time slice.

Table 5–20: Relational Subspaces for outliers from $X_1[t]$ =Conductivity analyzer.

Outliers	**DSTAPs**	**Time slices**
Global points	$(X_3[t] = 2) \rightarrow (X_1[t] = 1)$	$t = 780 : 784, 890 : 894, 1590 : 1594$
	$\Big[(X_1[t] = 1), (X_3[t+1] = 2)\Big] \rightarrow (X_1[t+1] = 1)$	$t = 610 : 613$
	$\Big[(X_1[t] = 5), (X_3[t+1] = 3)\Big] \rightarrow (X_1[t+1] = 1)$	$t = 609$

The reported interesting temporal outliers from time series $X_2[t]$ pH analyzer that measures the HCl level is associated with the DSTAP 3. Table 5–21shows the relational subspaces which were targeted by interesting temporal outliers for $X_2[t]$ presented in the SWaT dataset. For example:

In time slices $t = 1310 : 1320$ the relational subspace:

$\big(Flow\ transmitter[t] = [1.9, 2.7)\big) \rightarrow \big(pH\ analyzer[t] = [9.02, 9.14)\big)$

Reports 11 interesting temporal outliers (contextual collective), on the pH analyzer with levels of HCl between $[9.02, 9.14)$, moreover, the probable anomaly cause is the Flow transmitter with leves of dosing pumps between $[1.9, 2.7)$.

Table 5–21: Relational Subspaces for outliers from $X_2[t]$ =pH analyzer.

Outliers	**DSTAP**	**Time slices**
Contextual collective	$\big(X_4[t] = 3\big) \rightarrow \big(X_2[t] = 7\big)$	$t = 1310 : 1320$

The reported interesting temporal outliers from time series $X_3[t]$ ORP analyzer that measures the NaOCl level is associated with the DSTAPs 1, 2, 7, and 8. Table 5–22 shows the relational subspaces which were targeted by interesting temporal outliers for $X_3[t]$ presented in the SWaT dataset. For example

In time slices $t = 610 : 620$ the relational subspace:

$\big(Flow\ transmitter[t] = [2.7, 3.6)\big) \rightarrow \big(ORP\ analyzer[t] = [524, 543)\big)$

Reports 11 interesting temporal outliers (contextual collective), on the ORP analyzer with levels of NaOCl between $[524, 543)$, moreover, the probable anomaly cause is the Flow transmitter with leves of dosing pumps between $[2.7, 3.6)$.

Table 5–22: Relational Subspaces for outliers from $X_3[t]$ =ORP analyzer.

Outliers	**DSTAPs**	**Time slices**
Contextual	$\big(X_4[t] = 4\big) \rightarrow \big(X_3[t] = 7\big)$	$t = 610 : 620, 1700 : 1704$
points	$\big(X_4[t] = 3\big) \rightarrow \big(X_3[t] = 7\big)$	$t = 1500 : 1510, 1668 : 1672, 1858 : 1862$
and	$\Big[\big(X_3[t] = 7\big), \big(X_4[t+1] = 4\big)\Big] \rightarrow \big(X_3[t+1] = 7\big)$	$t = 611 : 619, 1701 : 1703$
collective	$\Big[\big(X_3[t] = 2\big), \big(X_4[t+1] = 3\big)\Big] \rightarrow \big(X_3[t+1] = 7\big)$	$t = 1667, 1857$

The reported interesting temporal outliers from time series $X_4[t]$ flow transmitter that control dosing pumps is associated with the DSTAP 5 and 6. Table 5–23 shows the relational subspaces which were targeted by interesting temporal outliers for $X_4[t]$ presented in the SWaT dataset. For example:

In time slices $t = 610 : 619$ the relational subspace:

$\big(Flow\ transmitter[t] = [2.7, 3.6)\big) \rightarrow \big(Flow\ transmitter[t+1] = [2.7, 3.6)\big)$

Reports 10 interesting temporal outliers (contextual collective), on the Flow transmitter with leves of dosing pumps between $[2.7, 3.6)$, moreover, the probable anomaly cause is the Flow transmitter with leves of dosing pumps between $[2.7, 3.6)$ in the previous time slice.

Table 5–23: Relational Subspaces for outliers from $X_4[t]$ =flow transmitter.

Outliers	**DSTAPs**	**Time slices**
Contextual	$\big(X_4[t] = 4\big) \rightarrow \big(X_4[t+1] = 4\big)$	$t = 610 : 619, 1480 : 1484, 1640 : 1644, 1670 : 1674$
points, collective	$\big(X_4[t] = 3\big) \rightarrow \big(X_4[t+1] = 4\big)$	$t = 609, 1479, 1669$

Efficiency Measures:

The experimental study on SWaT-dataset has been performed in a supervised mode; thus, the labels (normal or outlier) are available for each instance. When dealing with imbalanced classes, it is appropriate to report precision and recall as efficiency measures to guarantee the real detection performance of the DSTAP method avoiding to get in a high ratio of correctly classified. Table 5–24 shows the summary of efficiency measures achieved using DSTAP methodology on the SWaT dataset.

Table 5–24: Precision and Recall achieved using DSTAP on DBN_{SWaT}.

Data Set	**Precision**	**Recall**
SWaT	0.8295	0.8111

The research work described in [56] reported the average area under the ROC curve $AUC = 0.85$ over 50 runs. Despite the precision and recall are both less than AUC, our method does not lose any information in the discovery process and does not run 50 times (falling in an excessive tuning), moreover, provides explanations and probable causes of the reported temporal outliers.

5.5 Discussion

The DSTAP methodology has the aim to discover interesting temporal outliers and contextualize their anomalous essence. Based on the performed experimental study, we may conclude that by considering causal probabilistic relations in the feature space, we can reach significant results in the discovery process. It is important to remark that techniques based on transformation do not provide contextualization of the reported outliers due to the reduction in nature.

We extracted temporal anomalous patterns, which are examples of *low support* & *high confidence* or *high support* & *low confidence* events. The temporal anomalous patterns were then tested on datasets to discover interesting temporal outliers. We proved the credibility of our approach over a toy example, existing well-known dynamic Bayesian network models and their synthetic datasets, and real datasets concerned with the water treatment system in the scenario of a cyber-physical system.

A detailed procedure of DSTAP has been performed in a toy example. The discovery process was done in an unsupervised scenario. The results have been shown that DSTAP can detect interesting temporal outliers and can provide an explanation about possible causes. The experimental study based on synthetic datasets was also performed in an unsupervised scenario. The experimental study was run on four dynamic Bayesian networks with different configurations. We can conclude that there exist computational limitations in the learning process of a DBN. Besides

the computational constraints in the learning process, we can discover meaningful temporal outliers and its relational subspaces to provide explanations.

To compare the performance of DSTAP with a reduction-based method, we employed the DSTAP method under the same configurations in [56]. We used the water treatment system (SWaT) datasets. A comparative experimental study has been done in a supervised scenario. To discover interesting temporal outliers in multidimensional time series. First, the DSTAP has been learned DBN_{SWaT} from datasets, then, DSTAP has been discovered outliers and their relational subspaces to provide explainability to the reported outliers. Our method presents excellent performance with the same results as in [56]. However, we can provide contextualization of the reported outliers; thus, we can enrich our knowledge with more information about the anomaly cause.

Chapter 6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This dissertation proposes a novel technique about discovering interesting temporal outliers and explains the interestingness of the reported outlier in temporal datasets. The previous works have been concentrated on mining temporal anomalous patterns. However, not providing a contextualization of the interestingness of the reported outlier. Most of these approaches are based on the transformation and dimensionality reduction of the temporal datasets, losing valuable information in the discovery process. The significant distinction of this research work is that it offers the ability to discover interesting temporal outliers and the relational subspaces where they appear, providing contextual information of the reported outliers in an effective and automated manner. The integration of domain knowledge has been needed in the discovery process. Dynamic Bayesian networks can capture and represent domain specific knowledge through the available temporal datasets. The process of structure and parameter learning in the dynamic Bayesian networks provide causal probabilistic relationships and a degree of belief within and between attributes that exist in the domain. The learning process in DBN guarantees the representation of the domain knowledge of the datasets. By taking advantage of these probabilistic relationships between attributes. The two probabilistic association rules were proposed as *low support & high confidence*, and *high support & low confidence*. These rules were used in order to discover de temporal anomalous patterns, and provide contextualization of the reported outlier through relational subspaces over a specific timestamp. This point of view to discover temporal outliers contradicts the

probabilistic causal semantic represented via a dynamic Bayesian network model. We proposed a novel methodology called "Domain Specific Temporal Anomalous Patterns," an algorithm and its implementation to discover and explain interesting temporal outliers. The experimentation has been done on discrete multivariate sequences and multivariate time series in synthetic and real datasets on unsupervised and supervised scenarios. The experimental results show that our approach can detect interesting temporal outliers and provide an explanation in the form of relational subspaces about the probable causes of the reported outliers, with reasonable efficiency measures, precision, and recall.

6.2 Future Work

This research has proposed a powerful technique to discover interesting temporal outliers using dynamic Bayesian networks and probabilistic association rules. However, there are some promising research directions to improve and extend the work presented in this thesis.

- The DSTAP methodology proposed aims to discover interesting temporal outliers and provide a contextualization of the reported outlier for discrete sequences since the algorithm to learn the DBN has been implemented for discrete random processes. In the case of time series, we performed discretization as a pre-processing step, in order to use the DSTAP methodology. However, we can extend the learning process in DBNs to Gaussian random process, called dynamic Gaussian Bayesian networks. In line with this thesis, we see a promising extension to discover interesting temporal outliers in time series without discretization as a pre-processing step.
- The assumptions of stationarity and first-order Markovian to learning the DBN model may be relaxed. Opening the problem to learn from a Non-Stationary Dynamic Bayesian Network of k-order Markovian dataset. At first impression, the

problem with these models may be more complicated than the proposed methodology. However, we see promising and more precise results over complex situations where the temporal dependence is not stationary and is not first-order Markovian.

- The DSTAP methodology was implemented in a sequential process, however, we see that our algorithm can be parallelized in the task scenario, specifically, in the computation of relational subspaces and applying the rules to each anomalous patterns. Data parallelism is not feasible because of the dependency structure in temporal datasets.

Chapter 7 ETHICAL CONSIDERATIONS

Ethical issues form an essential component of modern research, related to the subject and researcher. Research ethics require the application of fundamental ethical principles in scientific research. The ethical principles include honesty, objectivity, integrity, carefulness, openness, responsibility, confidentiality, legality, and respect for intellectual property, all of these based on the *Nuremberg Code* and the *Declaration of Helsinki* [57], represent relevant literature to the ethical and legal aspects of conducting research.

In the Data Mining community, the design and implementation of methods and algorithms poses an ethical issue covering privacy, data accuracy, database security, reproducibility, stereotyping, legal liability, and the broader research dilemmas [58]. In Data Mining, data-driven processes represent an ethical issue in the scenario of decision making affect people or compromises their privacy [59].

The ethical aspects in Cyber-Physical systems [60] have been responsibly fulfilled, due to the experimental use of the secure water treatment (SWaT) system dataset in this thesis. The SWaT dataset corresponded to a Cyber-Physical system, and was declared by the authors as open access for research in the anomaly detection community.

The area of this thesis is Data Mining. The ethical principles mentioned above were considered thoroughly in this thesis. The proposed methodology has been developed and implemented fulfilling ethical principles. The temporal datasets used in this thesis have been described clearly and are declared as general use and open access. The experimental results have been discussed objectively, without any influence at all, to arrive at a particular conclusion.The implementation and results of this thesis have been performed openly to ensure the reproducibility of this thesis in scientific research, especially in the Data Mining community. In this thesis, no sensitive data has been used, e.i., datasets regarding experimentation on humans or other living species.

Bibliography

[1] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[2] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 139–147. Morgan Kaufmann Publishers Inc., 1998.

[3] D. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[4] C. Aggarwal. *Outlier analysis*. Springer Science & Business Media, 2013.

[5] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct 2004.

[6] S. Babbar, D. Surian, and S. Chawla. A causal approach for mining interesting anomalies. In *Advances in Artificial Intelligence*, pages 226–232. Springer, 2013.

[7] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5(1):1–129, 2014.

[8] V. Chandola. *Anomaly detection for symbolic sequences and time series data*. PhD thesis, University of Minnesota, Minnesota, 2009.

[9] S. Babbar and S. Chawla. On bayesian network and outlier detection. In *COMAD*, page 125, 2010.

[10] K. P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[11] I. Melnyk. *Dynamic Bayesian Networks: Estimation, Inference and Applications*. PhD thesis, University of Minnesota. Minnesota, 2016.

[12] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[13] S. Babbar. *Inferring Anomalies from Data using Bayesian Networks*. PhD thesis, University of Sydney. Graduate School of Engineering IT School of Information Technologies., 2013.

[14] L. Geng and H. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9, 2006.

[15] A. Masood. *Measuring Interestingness in Outliers with Explanation Facility using Belief Networks*. PhD thesis, Nova Southeastern University. Graduate School of Computer and Information Sciences., 2014.

[16] Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 12(2):159–170, 2010.

[17] A. Cansado and A. Soto. Unsupervised anomaly detection in large databases using bayesian networks. *Applied Artificial Intelligence*, 22(4):309–330, 2008.

[18] D. Barbara, N. Wu, and S.Jajodia. Detecting novel network intrusions using bayes estimators. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–17, 2001.

[19] D. Janakiram, A. Mallikarjuna, V. Reddy, and P. Kumar. Outlier detection in wireless sensor networks using bayesian belief networks. In *Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on*, pages 1–6. IEEE, 2006.

[20] W.K. Wong, A. Moore, G. Cooper, and M. Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *ICML*, pages 808–815, 2003.

[21] S. Babbar and S. Chawla. Mining causal outliers using gaussian bayesian networks. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, pages 97–104, Nov 2012.

[22] S. Babbar. Detecting and describing non-trivial outliers using bayesian networks. In *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, pages 1–6, March 2015.

[23] A. Masood and W. Li. Finding interesting outliers - a belief network based approach. In *SoutheastCon 2015*, pages 1–7, April 2015.

[24] S. Babbar. Integration of domain knowledge for outlier detection in high dimensional space. In *Database Systems for Advanced Applications*, pages 363–368, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[25] T. Mitsa. *Temporal Data Mining*. Chapman & Hall/CRC, 1st edition, 2010.

[26] R. Baragona and F. Battaglia. Outliers detection in multivariate time series by independent component analysis. *Neural Comput.*, 19(7):1962–1984, July 2007.

[27] R. S. Tsay, D. Peña, and P. Galeano. Outlier detection in multivariate time series via projection pursuit. Working papers. statistics and econometrics., Universidad Carlos III de Madrid. Departamento de Estadstica, 2004.

[28] H. Cheng, P. Tan, C. Potter, and S. Klooster. A robust graph-based algorithm for detection and characterization of anomalies in noisy multivariate time series. In *2008 IEEE International Conference on Data Mining Workshops*, pages 349–358, Dec 2008.

[29] D. Hill, B. S Minsker, and E. Amir. Real-time bayesian anomaly detection for environmental sensor data. In *Proceedings of the Congress-International Association for Hydraulic Research*, volume 32, page 503. Citeseer, 2007.

[30] A. Ogbechie, J. Díaz-Rozo, P. Larrañaga, and C. Bielza. Dynamic bayesian network-based anomaly detection for in-process visual inspection of laser surface heat treatment. In *Machine Learning for Cyber Physical Systems*, pages 17–24. Springer, 2017.

[31] M. Saada, Q. Meng, and T. Huang. A novel approach for pilot error detection using dynamic bayesian networks. *Cognitive neurodynamics*, 8(3):227–238,

2014.

[32] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

[33] D. Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354. Springer, 1998.

[34] R. Daly, Q. Shen, and S. Aitken. Learning bayesian networks: approaches and issues. *The Knowledge Engineering Review*, 26(2):99–157, 2011.

[35] R. Nagarajan, M. Scutari, and S. Lèbre. *Bayesian Networks in R*. Springer, 2013.

[36] D. Margaritis. *Learning Bayesian network model structure from data.* PhD thesis, Computer Science, Carnegie-Mellon University, Pittsburgh, PA., 2003.

[37] N. Friedman, I. Nachman, and D. Peér. Learning bayesian network structure from massive datasets: the sparse candidate algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc., 1999.

[38] I. Tsamardinos, L. E. Brown, and C. F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[39] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[40] Z. Ghahramani. Learning dynamic bayesian networks. In *Adaptive processing of sequences and data structures*, pages 168–197. Springer, 1998.

[41] V Mihajlovic and M Petkovic. Dynamic bayesian networks: A state of the art. *University of Twente Document Repository*, 2001.

[42] J. W. Robinson and A. J. Hartemink. Learning non-stationary dynamic bayesian networks. *Journal of Machine Learning Research*, 11:3647–3680, December 2010.

[43] G. Trabelsi, P. Leray, M. Ben Ayed, and A. M. Alimi. Benchmarking dynamic bayesian network structure learning algorithms. In *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, pages 1–6, April 2013.

[44] S. Gao, Q. Xiao, Q. Pan, and Q. Li. Learning dynamic bayesian networks structure based on bayesian optimization algorithm. In *International Symposium on Neural Networks*, pages 424–431. Springer, 2007.

[45] H. Wang and H. Yu, K.and Yao. Learning dynamic bayesian networks using evolutionary mcmc. In *Computational Intelligence and Security, 2006 International Conference on*, volume 1, pages 45–50. IEEE, 2006.

[46] J. M. Peña, J. Björkegren, and J. Tegnér. Learning dynamic bayesian network models via cross-validation. *Pattern Recognition Letters*, 26(14):2295–2308, 2005.

[47] Y. Lou, Y. Dong, and H. Ao. Structure learning algorithm of dbn based on particle swarm optimization. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 102–105. IEEE, 2015.

[48] G. Trabelsi, P. Leray, M. B. Ayed, and A. M. Alimi. Dynamic mmhc: A local search algorithm for dynamic bayesian network structure learning. In *International Symposium on Intelligent Data Analysis*, pages 392–403. Springer, 2013.

[49] P Chaudhari, Dipti P Rana, Rupa G Mehta, NJ Mistry, and Mukesh M Raghuwanshi. Discretization of temporal data: a survey. *International Journal of Computer Science and Information Security 11 (2), 66-69*, 2014.

[50] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[51] G. Trabelsi. *New structure learning algorithms and evaluation methods for large dynamic Bayesian networks*. PhD thesis, Université de Nantes; Ecole Nationale d'Ingénieurs de Sfax, 2013.

[52] H. Chan and A. Darwiche. Sensitivity analysis in bayesian networks: From single to multiple parameters. *arXiv preprint arXiv:1207.4124*, 2012.

[53] T. Griffiths and J. Tenenbaum. From mere coincidences to meaningful discoveries. *Cognition*, 103(2):180–226, 2007.

[54] K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33:2001, 2001.

[55] J. Goh, S. Adepu, K. Junejo, and A. Mathur. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security*, pages 88–99. Springer, 2016.

[56] M. Hulsebos. *Outlier detection in multivariate time series: exploiting reconstructions from random projections.* Master's thesis, Pattern Recognition Laboratory of Delft University of Technology., Netherlands, 2018.

[57] J. Kruk. Good scientific practice and ethical principles in scientific research and higher education. *Central European Journal of Sport Sciences and Medicine*, 2013.

[58] K. Wahlstrom, J. Roddick, R. Sarre, V. Estivill-Castro, and D. deVries. On the ethical and legal implications of data mining. *Technical Report SIE-06-001,University of Adelaide, Australia.*, 2006.

[59] P. Fule and J. Roddick. Detecting privacy and ethical sensitivity in data mining results. In *Proceedings of the 27th Australasian conference on Computer science-Volume 26*, pages 159–166. Australian Computer Society, Inc., 2004.

[60] A. Thekkilakattil and G. Dodig-Crnkovic. Ethics aspects of embedded and cyber-physical systems. In *IEEE 39th Annual Computer Software and Applications Conference*, 2015.