2018

Exploring Granger causality in dynamical systems modeling and performance monitoring

Homagni Saha
Iowa State University

Exploring Granger causality in dynamical systems modeling and performance monitoring

by

Homagni Saha

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee:
Soumik Sarkar, Major Professor
Chinmay Hegde
Sourabh Bhattacharya

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

DEDICATION

I would like to dedicate this thesis to my parents Sumana Saha and Asit Kumar Saha without whose support I would not have been able to complete this work. I would also like to thank my friends and family for their loving guidance and financial assistance during the writing of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Soumik Sarkar for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Chinmay Hegde and Dr. Sourabh Bhattacharya.

ABSTRACT

Data-driven approaches are becoming increasingly crucial for modeling and performance monitoring of complex dynamical systems. Such necessity stems from complex interactions among sub-systems and high dimensionality that render majority of first-principle based methods insufficient. This work explores the capability of a recently proposed probabilistic graphical modeling technique called spatiotemporal pattern network (STPN) in capturing Granger causality among observations in a dynamical system. In this context, we introduce the notion of Granger-STPN (G-STPN) inspired by the notion of Granger causality. We compare the metrics used in the two frameworks for increasing memory in a dynamical system, and show that the metric for G-STPN can be approximated by transfer entropy. We apply this new framework for anomaly detection and root cause analysis in a robotic platform.

CHAPTER 1. OVERVIEW

Large-scale cyber-physical systems (CPSs) are being explored widely in various application sectors, e.g., transportation networks (1), integrated buildings (2), robotic networks (3), wind farms (4), and smart home Internet of Things (IoT) (5). In such systems, the interactions between different parts or subsystems are critically important for the purposes of control and decision-making. While physics-based methods model such interactions using first principles, it becomes significantly complicated as the number of subsystems increases along with their complex interactions. Therefore, data-driven methods have been receiving considerable attention from the industry and academia (6; 7) as they tend to be more scalable and accurate. However, modeling spatiotemporal causal interactions is non-trivial and crucial for the sake of performance monitoring and diagnose issues as well as developing advanced control techniques. For example, information theoretic measures such as Granger Causality can give relevant insights when considering the effectiveness of control mechanisms (8). Although identifying such causal relations has been explored in neuroscience (9), finance (10), and even social sciences (11), the applications to large-scale cyber-physical systems have not been explored sufficiently.

Recently, a probabilistic graphical modeling technique called spatiotemporal pattern network (STPN) has been shown to be quite effective in modeling distributed cyber-physical systems using multivariate time series observation from the system. Built upon the concepts of Symbolic Dynamic Filtering (12), this data-driven framework has been used in a variety of applications to diagnose and predict system behavior. For example, it has been used for prediction of wind energy (13), residential energy disaggregation (2), and root-cause analysis of physical faults and cyber anomalies in CPSs (14). Although STPN attempts to capture relational patterns among different observations or sub-systems using information-theoretic measures, a rigorous causality analysis hasn't been performed.

This work explores the capability of STPN in capturing Granger causality among observations in a dynamical system. In this context, we introduce a variant of STPN namely, Granger-STPN (G-STPN) that leverages the concept of transfer entropy computed between two symbolic time series that can capture Granger causality. The key difference between STPN and G-STPN is that G-STPN needs to consider the product state space of the two symbolic time series as opposed to individual state space consideration in STPN. Therefore, G-STPN may suffer from scalability issues while considering more memory or longer history for the symbolic time series observations. However, in some systems even increasing the memory of system under consideration for joint states, does not significantly increase dimensions of the product space. Even so, we borrow a state merging approach from previous work to apply G-STPN to a practical problem of anomaly detection involving an industrial robotic platform. For single node anomalies, our framework is able to isolate the causes of anomaly using root cause analysis approach.

Contributions

1. Extend the recently proposed STPN framework as a non-linear model of granger causality. We call it Granger STPN (G-STPN).

2. Use G-STPN for performance monitoring of an industrial manipulator by using anomaly detection and root cause analysis approaches.

CHAPTER 2. PRELIMINARY CONCEPTS

2.1 Preliminaries

2.1.1 Formulation of Granger causality

In this section, we will briefly discuss the concept and formulation of Granger causality. In the simplest form, the core idea of Granger causality is that a process X Granger causes Y if it helps in increasing the prediction power of a model based on the past history of X and Y than that based on the past history of Y alone. We will go through the essentials in brief. Let $F(x_t|x_{t-1}^k, y_{t-1}^k)$ denote the distribution function of the variable X (call it target variable), conditional on the joint (k-lag) history x_{t-1}^k, y_{t-1}^k of both itself and the variable Y (call it source variable), and let $F(x_t|x_{t-1}^k)$ denote the distribution function of X_t conditional on just its own k-history. Then variable Y is said to Granger cause variable X (8) (with k lags) iff:

$$F(x_t|x_{t-1}^k, y_{t-1}^k) \neq F(x_t|x_{t-1}^k) \tag{2.1}$$

Granger's formulation was based on vector autoregressive modeling (VAR). Let X_t and Y_t be two multivariate real-valued, zero-mean, jointly stationary stochastic processes. Consider the nested VAR models:

$$X_t = \sum_{i=1}^{k} A_i X_{t-i} + \sum_{i=1}^{k} B_i Y_{t-i} + \epsilon_t \tag{2.2}$$

$$X_t = \sum_{i=1}^{k} \tilde{A}_i X_{t-i} + \tilde{\epsilon}_t \tag{2.3}$$

The model has parameters A_i, B_i, $\tilde{A}_i$ and the covariance matrices $\Sigma(\epsilon_t)$ (we will denote as Σ), $\Sigma(\tilde{\epsilon}_t)$ (we will denote as $\tilde{\Sigma}$). Here ϵ_t, $\tilde{\epsilon}_t$ are the residuals assumed to be serially uncorrelated. Equations 2.2 and 2.3 denote the full and reduced models respectively. Granger, in his original approach views equations 2.2 and 2.3 as predictive models for the target variable X in terms of the joint past history of X and Y. Accordingly, the $Y \longrightarrow X$ Granger causality statistic attempts to quantify the degree to which the full model yields a better prediction of the target variable than the reduced model. Thus, Granger causality from Y to X it is given by:

$$F^{k}_{Y \longrightarrow X} \equiv log(\frac{|\tilde{\Sigma}|}{|\Sigma|}) \tag{2.4}$$

Where $|.|$ denotes matrix determinant. The determinant of covariance matrix of residuals is also known as generalized variance. Granger causality is linear and parametric by definition.

2.1.2 Formulation of transfer entropy

This section introduces the preliminaries for another non parametric and non linear causality metric - transfer entropy. We first state the Shannon entropy and Kullback entropy, which are given by the following expression, respectively:

$$H_I = \sum_{n} p(i) log(p(i)) \tag{2.5}$$

$$K_I = \sum_{n} p(i) log\left(\frac{p(i)}{q(i)}\right) \tag{2.6}$$

where $i = 1, 2, ..., n$, for all states (total of n) the process I can assume, $p(i)$ and $q(i)$ are the probability distribution. The definition of Kullback divergence can be extended for a process having a joint distribution $p(i, j)$ which is erroneously assumed to have been enumerated by two independent distributions $p(i)$ and $p(j)$ to define mutual information between two processes I and J generating observations $\{i\}$ and $\{j\}$ respectively.

$$M^{IJ} = \sum_{m,n} p(i, j) log\left(\frac{p(i, j)}{p(i)p(j)}\right) \tag{2.7}$$

Mutual information can be quantified in terms of entropy as

$$M^{IJ} = H_I + H_J - H_{IJ}, \tag{2.8}$$

which is always greater than 0 and is symmetric with respect to I and J. This prevents observing "information flows" from one subsystem to another. Also, there is no sense of "history of observed data" involved in this formulation. Transfer Entropy tries to deal with this issue. In the context of information theory, we convert continuous time domain signal into a discretized symbol sequence. In this work we use the Symbolic Dynamic Filtering technique (12) to obtain this. We first give the conditional entropy as follows:

$$H_{I|J} = \sum_{n} p(i,j) log(p(i|j)) = H_{IJ} - H_J \tag{2.9}$$

We consider a length of history (k), and systems I and J generating i_{n+1}, $i_n^k := \{i_n, i_{n-1}, ..., i_{n-k}\}$ (the k lag history of the sequence I) and $j_n^k := \{j_n, j_{n-1}, ..., j_{n-k}\}$.(the k lag history of the sequence J). According to (15) transfer entropy for two systems is defined as the difference of conditional entropies as follows:

$$\mathcal{T}_{I\rightarrow J} = H_{J|J^-} - H_{J|J^- I^-} \tag{2.10}$$

where I and J correspond to i_{n+1} and j_{n+1}, and I^- and J^- are the histories of processes I and J, corresponding to i_n^k and j_n^k respectively. There exists efficient methods in literature to calculate both the values of conditional entropies as given by the following equations

$$H(I_n|I_n^k, J_n^k) = -\mathbb{E}[logp(I_n|I_n^k, J_n^k)] \tag{2.11}$$

$$H(I_n|I_n^k) = -\mathbb{E}[logp(I_n|I_n^k)] \tag{2.12}$$

On application of equation 2.9 to equation 2.10, and substituting we find that:

$$\begin{aligned} \mathcal{T}_{I\rightarrow J} = \sum p(j_{n+1}, j_n^k, i_n^k) log(j_{n+1}|j_n^k, i_n^k) \\ - \sum p(j_{n+1}, j_n^k) log(j_{n+1}|j_n^k) \end{aligned} \tag{2.13}$$

There is a significance of k in the numerical form of transfer entropy. It controls how deep we investigate the history of both the variables. If a Markov chain of order k is assumed to generate these observations, we have:

$$p(j_{n+1}, j_n^k) = p(j_{n+1}, j_n^k, i_n^k) \tag{2.14}$$

In reality, the Markov assumption may not hold. Therefore, according to the Kullback-Liebler divergence from the generalized Markov assumption (16) we have :

$$\mathcal{T}_{I \to J} = \sum_{j_{n+1}, j_n^k, i_n^k} p(j_{n+1}, j_n^k, i_n^k) log\left(\frac{p(j_{n+1}|j_n^k, i_n^k)}{p(j_{n+1}|j_n^k)}\right) \tag{2.15}$$

We consider this in a symbolic domain, thus it is an expression for symbolic transfer entropy, which is elaborated in detail in (17). Equation 2.15 can be calculated in a more efficient manner (although approximately in the symbolic domain) using equation 2.10. In (18) the authors have shown that the transfer entropy is equivalent to Granger causality for Gaussian variables.

CHAPTER 3. GRANGER-STPN FRAMEWORK

In this section we extend the STPN to Granger-STPN (G-STPN). First, the STPN framework is revisited. For more details, please see (13; 2).

3.1 STPN

The STPN is a Graph, the nodes of which are defined using xD-Markov machines (13). An xD-Markov machine is a five tuple which signifies the state transition matrices between different systems. An STPN is intended to portray strengths of "causality" between different nodes of the graph which correspond to the variables of a multivariate time series under observation. We give the definition of STPN as follows:

Definition 3.1.1 *A STPN is a 4-tuple* $W = (Q^I, \Sigma^J, \Pi^{IJ}, M^{IJ})$*:(*$I, J$ *denote nodes of the STPN)*

1. $Q^I = \{q_1, q_2, , q_{|Q_I|}\}$ *is the state set corresponding to symbol sequences* S^I*;*

2. $\Sigma^J = \{\sigma_0, ..., \sigma_{|\Sigma^J|-1}\}$ *is the alphabet set of symbol sequence* S^J*;*

3. Π^{IJ} *is the symbol generation matrix of size* $|Q^I| \times |\Sigma^J|$*, the* ij^{th} *element of* Π^{IJ} *denotes the probability of finding the symbol* σ_j *in the symbol string* S^J *while making a transition from the state* q_i *in the symbol sequence* S^I*; while self-symbol generation matrices are called atomic patterns (APs) i.e.,* $I = J$*, cross-symbol generation matrices are called relational patterns (RPs) i.e., when* $I \neq J$*.*

4. M^{IJ} *denotes a metric that can represent the importance of the learnt pattern (or degree of causality) for* $I \rightarrow J$ *which is a function of* Π^{IJ}*.*

A visualization of STPN is provided in figure 3.1. As defined above, we learn two types patterns in an STPN, namely relational pattern (RP) to capture the capability of a sequence in predicting another sequence and atomic pattern (AP) to capture self-prediction capability of a symbol sequence. Recall the definition of Granger-causality (19), a time evolving variable I Granger-causes a time-evolving variable J if predictions of J based on J's past values and on I's past values are better than J's predictions considering only its own past values. Therefore, to identify whether a sequence Granger-causes another in the context of STPN, we need to compare the information contents of AP and RP as well as examine how they jointly enable the prediction of the target sequence. However, as STPN only uses individual state spaces to describe AP and RP, it may be difficult to compare them without rigorous normalization processes and to compute the joint prediction capability. Therefore, we consider a variant of the STPN framework, called Granger-STPN (G-STPN) that considers the joint/product state space. In this case, the relational patterns in STPN get replaced by the joint patterns as shown in figure 3.1. We rigorously define Granger-STPN (G-STPN) in the sequel.

3.2 G-STPN framework

In this section, We introduce the definition of Granger-STPN framework as shown in figure 3.1. The Granger-STPN follows the definition of STPN while changes are primarily made to the part (3) of the definition. Formally, we have

Definition 3.2.1 *A Granger-STPN is a 6-tuple $W = (Q^I, Q^J, \Sigma^J, \tau^{IJ}, \Pi^{IJ}, T^{IJ})$: ($I, J$ denote nodes of the Granger-STPN which are basically two different variables of a multivariate time series where causality is investigated)*

1. *$Q^I = \{q_1, q_2, ..., q_{|Q_I|}\}$ is the state set corresponding to all k-lag embeddings of symbol sequences S^I;*

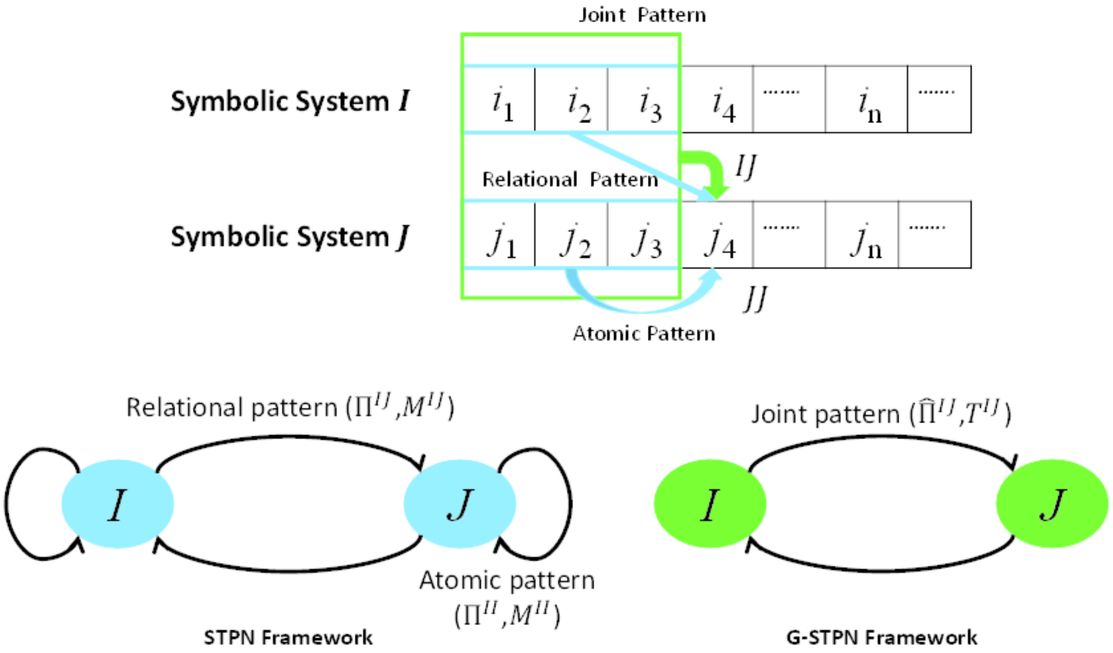


Figure 3.1 STPN framework (blue)and G-STPN framework (green) involving systems I and J

2. *$Q^J = \{q_1, q_2, ..., q_{|Q_J|}\}$ is the state set corresponding to all k-lag embeddings of symbol sequences S^J;*

3. *$\Sigma^J = \{\sigma_0, ..., \sigma_{|\Sigma^J|-1}\}$ is the alphabet set of symbol sequence S^J;*

4. *τ^{IJ} is the joint state-symbol generation matrix of size $|Q^I| \times |Q^J| \times |\Sigma^J|$, the ij^{th} element of τ^{IJ} denotes the probability of finding the symbol σ_j in the symbol string S^J while making a transition from the state $q_i \in Q^I$ and (jointly) state $q_j \in Q^J$ such a pattern is called **joint pattern** between nodes of I and J;*

5. *Π^J is the self state symbol generation matrix of size $|Q^J| \times |\Sigma^J|$, the ij^{th} element of Π^J denotes the probability of finding the symbol σ_j in the symbol string S^J while making a transition from the state $q_j \in Q^J$. Such a pattern is called **atomic pattern** in node J. Alternatively, it can be viewed as a parameter matrix that can be used to calculate **active information storage (AIS)**.*

6. *T^{IJ} denotes a metric that can represent causality (degree of influence of variation of J on I), denoted $I \to J$ which is a function of τ^{IJ}, and here we define it as a log likelihood ratio test statistic of a short time window realization of the joint process (I, J).*

Remark 3.2.1 *Based on Section 2.1 it can be observed that the difference between STPN and G-STPN is whether the history of one process itself is taken into account when using another process to predict it. Equivalently speaking, in STPN, the relational patterns between different systems are considered solely without the atomic patterns. Intuitively, the G-STPN should be more accurate for capturing the causal relationship between the systems. It can be suggested that STPN approximates G-STPN under some special conditions. It is evident that the importance metric of STPN is susceptible to common cause effects which may interfere with causal inference, but the importance metric of G-STPN is, as we will show shortly, very closely related to transfer entropy, which enables it to get around such effects by separating past histories.*

3.3 Joint state merging algorithm

While modeling the joint state symbol generation matrix we would require the row vectors of τ^{xy} to be statistically independent of each other. Also we would require a mechanism to prevent the dimensions of the matrix τ^{xy} from increasing dramatically with increasing number states in Q^I, Q^J and alphabets in Σ^J. Therefore we use the state merging algorithm which is covered in detail in (2). Here we would explain the main steps of the algorithm in brief. Following from the definition of G-STPN above, here σ_n^J denotes the n^{th} symbol observed in system J, and q_r^{IJ} denotes the r^{th} joint state observed in the joint state space of the system I and J. For a multivariate time series we would have to calculate parameters of G-STPN for each pair of time series $x \in I$ and $y \in J$, in the entire system. We f, which is the total number of time series present in the system. The first step is to evaluate a metric

$\gamma(r)$ which would give us the importance of the r^{th} state as given in the equation below:

$$\gamma(r) = \|Pr(q_r^{IJ}, \sigma_n^J) - \bar{Pr}(q_r^{IJ}, \sigma^J)\|_1, r = 1, 2, ..., \prod_{x=1}^{f} \prod_{y=1; y \neq x}^{f} |Q^x| \times |Q^y| \tag{3.1}$$

where

$$\bar{Pr}(q_r^{IJ}, \sigma^J) = \sum_{n=1}^{\prod_{y=1}^{f} |\sigma^y|} \frac{Pr(q_r^{IJ}, \sigma_n^J)}{\prod_{y=1}^{f} |\sigma^y|} \tag{3.2}$$

If $\gamma(r) < \eta$ where η is a specified threshold, the state q_r^{IJ} is identified to be merged to other states. Then the relevance $\Gamma(r, s)$ of the two states is defined as:

$$\begin{aligned} \Gamma(r, s) = & \sum_{n=1}^{\prod_{y=1}^{f} |\sigma^y|} \|Pr(q_r^{IJ}, \sigma_n^J) - Pr(q_s^{IJ}, \sigma_n^J)\|_1, \\ & r, s = 1, 2, ..., \prod_{x=1}^{f} \prod_{y=1; y \neq x}^{f} |Q^x| \times |Q^y| \end{aligned} \tag{3.3}$$

$\Gamma(r, s)$ can be applied to find out the closest state q_s^{IJ} to be merged where $\gamma(s) \geq \eta$. Also we can merge states q_r and q_s if $\Gamma(r, s) \leq \mathscr{D}$, where q_r^{IJ} and q_s^{IJ} are the states with very similar transition probabilities and $\mathscr{D}$ is a specified threshold.

3.4 Modeling the joint symbol generation matrix

In this section we will model the matrix τ^{xy} using the Dirichlet distribution, and find out an expression for its prior joint density conditioned on a realization (x_n, y_n) of n data-points from two variables of the time series X and Y. Recall that τ^{xy} depends on the k-lagged history of (x_n) that constitutes the state sequence Q_t^x, the k-lagged history of (y_n) that constitutes the state sequence Q_t^y, and the symbol sequence (y_n) which we denote as S_t^y to avoid ambiguity. Thus τ^{xy} is a structure with $|Q_t^x| \times |Q_t^y|$ rows and $|S_t^y|$ columns. **After performing state merging we proceed with the assumption that individual rows of τ^{xy} are statistically independent of each other**. Each row of τ^{xy} is treated as a random vector. For the mth row, a prior probability density function $g_{\tau_m^{xy} | Q_t^x, Q_t^y, S_{t+1}^y}$ for the random matrix τ^{xy} conditioned on the joint state-symbol sequence Q_t^x, Q_t^y, S_{t+1}^y (t denoting time index of the variable in observed sequence, from here on we wont be using the subscript t to make it simpler) following the Dirichlet distribution is described below:

$$g_{\tau_m^{xy}|Q_t^x,Q_t^y,S_{t+1}^y} = \frac{1}{B(\alpha_m^{xy})} \times \prod_{n=1}^{|\Sigma^y|} (\theta_{mn}^{xy})^{(\alpha_{mn}^{xy}-1)} \tag{3.4}$$

where θ_m^{xy} is a realization of the random vector τ_m^{xy}, namely $\theta_{mn}^{xy} = [\theta_{m1}^{xy}\theta_{m1}^{xy}...\theta_{m|\Sigma^y|}^{xy}]$, and the normalizing constant is given by

$$B(\alpha_m^{xy}) \approx \frac{\prod_{n=1}^{\Sigma^y} \Gamma(\alpha_{mn}^{xy})}{\Gamma(\sum_{n=1}^{\Sigma^y} \alpha_{mn}^{xy})} \tag{3.5}$$

where $\alpha_{mn}^{xy} \equiv [\alpha_{m1}^{xy}\alpha_{m2}^{xy}...\alpha_{m|\Sigma^y|}^{xy}]$ with $\alpha_{mn}^{xy} = N_{mn}^{xy} + 1$ and N_{mn}^{xy} is the number of times the symbol $\sigma_n^y \in \Sigma^y$ is obtained after the joint state $q_m^{xy} \in |Q^x| \times |Q^y|$. It follows from equation 3.5 that

$$B(\alpha_m^{xy}) = \frac{\prod_{n=1}^{\Sigma^y} \Gamma(N_{mn}^{xy}+1)}{\Gamma(\sum_{n=1}^{\Sigma^y} N_{mn}^{xy} + |\Sigma^y|)} = \frac{\prod_{n=1}^{\Sigma^y} N_{mn}^{xy}!}{(N_{mn}^{xy} + |\Sigma^y| - 1)!} \tag{3.6}$$

Assuming the Markov property of the learned PFSA, the row vectors of τ^{xy} are statistically independent of each other. Therefore it follows from equations 3.4 and 3.6 that the prior joint density $g_{\tau_m^{xy}|Q_t^x,Q_t^y,S_{t+1}^y}$ of the joint state symbol generation matrix τ^{xy} conditioned on the joint state-symbol sequences Q_t^x, Q_t^y, S_{t+1}^y is given by

$$\begin{aligned} & g_{\tau^{xy}|Q_t^x,Q_t^y,S_{t+1}^y}(\theta^{xy}|Q_t^x, Q_t^y, S_{t+1}^y) \\ & = \prod_{m=1}^{|Q^x|\times|Q^y|} (N_m^{xy} + |\Sigma^y| - 1)! \prod_{n=1}^{|\Sigma^y|} \frac{(\theta_m^{xy})^{N_{mn}^{xy}}}{N_{mn}^{xy}!} \end{aligned} \tag{3.7}$$

where $\theta^{xy} = [(\theta_1^{xy})^T(\theta_2^{xy})^T...(\theta_{|Q^x|\times|Q^y|}^{xy})^T] \in [0,1]^{|Q^x|\times|Q^y|\times|\Sigma^x|}$.

3.5 Modeling self symbol generation matrix

In a way very similar to section 3.4, we will model a matrix Π^y using the Dirichlet distribution and find out an expression for its prior joint density conditioned on a realization (x_n) of n data points from the variable X of the time series. The matrix Π^y is the self state symbol generation matrix of size $|Q^Y|\times|\Sigma^Y|$, the ij^{th} element of Π^Y denotes the probability of finding the symbol σ_j in the symbol string S^Y while making a transition from the state $q_i \in Q^Y$. Such a pattern is called **atomic pattern** in node Y. Alternatively, it can be viewed as a parameter matrix that can be used to calculate **active information storage (AIS)**. Elements of the matrix Π^y are calculated by marginalizing elements of the matrix τ^{xy} over all possible states $q_x \in |Q^X|$. The prior probability density of the state symbol generation matrix Π^y conditioned on the self state-symbol sequences Q^y, S^y will be given by:

$$\begin{aligned} & h_{\Pi^y|Q_t^y,S_{t+1}^y}(\theta^{Yy}|Q_t^y,S_{t+1}^y) \\ & = \prod_{m=1}^{|Q^y|}(N_m^{Yy}+|\Sigma^y|-1)!\prod_{n=1}^{|\Sigma^y|}\frac{(\theta_m^{Yy})^{N_{mn}^{Yy}}}{N_{mn}^{Yy}!} \end{aligned} \tag{3.8}$$

where $\theta^{Yy} = [(\theta_1^{Yy})^T(\theta_2^{Yy})^T...(\theta_{|Q^y|}^{Yy})^T] \in [0,1]^{|Q^y|\times|\Sigma^y|}$. N_{mn}^{Yy} is the number of times the symbol $\sigma_n^y \in \Sigma^y$ is obtained after the state $q_m^y \in |Q^y|$. N_{mn}^{Yy} can be found out as follows:

$$N_{mn}^{Yy} = \sum_{k=1}^{k=|Q^X|} N_{mnk}^{xy} \tag{3.9}$$

3.6 Inferring about test observations

G-STPN is a pattern based framework that can capture multiple operating nodes of a system. A graphical model learned at an instant may not be same as the graphical model learned at another instant, provided the system moves on to a different mode of operation. However, this change can be recorded as patterns in the graph connectivity. The job of importance metric T defined in earlier sections is to represent these changes in flow of information from one node of the graphical model to another. However, there is no direct way to infer this change. One way to go about it is to evaluate the likelihood of a subsequence on data based on past observed data. We first model the nominal distribution of the data through τ^{xy}. During online inference we calculate the likelihood of an observed small time window of the data based on our prior model. We denote variables collected during inference phase with a tilde symbol in the superscript. For an observed subsequence of symbols in process Y, we denote the joint symbols of X and Y collected in training stage as $\mathcal{S}_\tau$, symbols of Y collected in training stage as $\mathcal{S}_\pi$ and those in inference stage as $\tilde{\mathcal{S}}$. We also use t and k to denote the time point of observed variables in the inference and training sequences respectively. We call X as the source time series and Y as the target time series. We are interested in determining the following two probabilities:

1. Probability that a Probabilistic Finite State Automaton (PFSA) with transition matrix τ^{xy} and joint state set of $|Q^X| \times |Q^Y|$ generated the subsequence $\tilde{\mathcal{S}}$. We call the model as full (joint) model, and denote the probability as Λ^{xy}.

2. Probability that a Probabilistic Finite State Automaton (PFSA) with transition matrix Π^y and a state set of $|Q^Y|$ generated the subsequence $\tilde{\mathcal{S}}$. We call the model as reduced (self) model, and denote the probability as λ^y.

We therefore define Λ and λ as:

$$\Lambda^{xy} = Pr(\tilde{Q}^x_t, \tilde{Q}^y_t, \tilde{S}^y_{t+1} | Q^x_k, Q^y_k, S^y_{k+1}) \equiv Pr(\tilde{\mathcal{S}} | \mathcal{S}_\tau) \tag{3.10}$$

$$\lambda^y = Pr(\tilde{Q}^y_t, \tilde{S}^y_{t+1} | Q^y_k, S^y_{k+1}) \equiv Pr(\tilde{\mathcal{S}} | \mathcal{S}_\pi) \tag{3.11}$$

We obtain a set of patterns Λ^{xy} and λ^y, for all x and y forming pairs of time series in the multivariate time series of the system. Let $\tilde{N}^{Yy}_{mn}$ denote the number of times in the short subsequence that the symbol σ^y_n was observed while there was a state $q_m \in |Q^Y|$. Then $\tilde{N}^{Yy}_m$ denotes the number of times the state $q_m \in |Q^Y|$ was observed in the short subsequence. Thus the probability that the self model generated $\mathcal{S}$ is given by the product of independent multinomial distributions.

$$Pr(\tilde{\mathcal{S}} | \Pi^y) = \prod_{m=1}^{|Q^y|} (\tilde{N}^{Yy}_m)! \prod_{n=1}^{|\Sigma^y|} \frac{(\Pi^y_{mn})^{\tilde{N}^{Yy}_{mn}}}{\tilde{N}^{Yy}_{mn}!} \tag{3.12}$$

The results from the testing is now conditioned on the training data. Given the symbol string $\mathcal{S}_\pi$ in the training phase the probability of observing the symbol string $\tilde{\mathcal{S}}$ is given by:

$$\begin{aligned} Pr(\tilde{\mathcal{S}}|\mathcal{S}_\pi) &= \int ... \int Pr(\tilde{\mathcal{S}}|\Pi^y = \theta^y) h_{\Pi^y|\mathcal{S}_\pi}(\theta^y|\mathcal{S}_\pi) d\theta^y \\ &= \int ... \int [\prod_{m=1}^{|Q^y|} (N_m^{\tilde{Y}y})! \prod_{n=1}^{|\Sigma^y|} \frac{(\theta_{mn}^y)^{\tilde{N}_{mn}^{Yy}}}{\tilde{N}_{mn}^{Yy}!}] \times \\ & [\prod_{m=1}^{|Q^y|} (N_m^{Yy} + |\Sigma^y| - 1)! \prod_{n=1}^{|\Sigma^y|} \frac{(\theta_m^{Yy})^{N_{mn}^{Yy}}}{N_{mn}^{Yy}!} d\theta_{mn}^y] \\ &= \prod_{m=1}^{|Q^Y|} (\tilde{N}_{mn}^{Yy})!(N_m^{Yy} + |\Sigma^Y| - 1)! \\ & \times \frac{\int ... \int \prod_{n=1}^{|\Sigma^Y|} {\theta_{mn}^{Yy}}^{\tilde{N}_{mn}^{Yy} + N_{mn}^{Yy}} d\theta_{mn}^y}{\prod_{n=1}^{|\Sigma^Y} (\tilde{N}_{mn}^{Yy})!(N_{mn}^{Yy})!} \end{aligned} \tag{3.13}$$

The integrand in equation 3.13 is the density function for the Dirichlet distribution up to the multiplication of a constant. Hence it follows from equation 3.6 that:

$$\int ... \int \prod_{n=1}^{|\Sigma^Y|} {\theta_{mn}^{Yy}}^{\tilde{N}_{mn}^{Yy} + N_{mn}^{Yy}} d\theta_{mn}^y = \frac{\prod_{n=1}^{|\Sigma^Y|} (\tilde{N}_{mn}^{Yy} + N_{mn}^{Yy})!}{(\tilde{N}_{mn}^{Yy} + N_{mn}^{Yy} + |\Sigma^Y| - 1)!} \tag{3.14}$$

Therefore, the probability of observing the symbol string $\tilde{\mathcal{S}}$ is given by:

$$\begin{aligned} Pr(\tilde{\mathcal{S}}|\mathcal{S}_\pi) = \lambda^y = \prod_{m=1}^{|Q^y|} & \frac{(\tilde{N}_m^{Yy})!(N_m^{Yy} + |\Sigma^y| - 1)!}{(\tilde{N}_m^{Yy} + N_m^{Yy} + |\Sigma^y| - 1)!} \\ & \prod_{n=1}^{|\Sigma^y|} \frac{(\tilde{N}_{mn}^{Yy} + N_{mn}^{Yy})!}{(\tilde{N}_{mn}^{Yy})!(N_{mn}^{Yy})!} \end{aligned} \tag{3.15}$$

The probability of the joint state-symbol subsequence is also a product of independent multinomial distributions given that the exact joint state symbol generation matrix is known.

$$Pr(\tilde{S}|\tau^{xy}) = \prod_{m=1}^{|Q^x| \times |Q^y|} (\tilde{N}_m^{xy})! \prod_{n=1}^{|\Sigma^y|} \frac{(\tau_{mn}^{xy})^{\tilde{N}_{mn}^{xy}}}{\tilde{N}_{mn}^{xy}!} \tag{3.16}$$

where, the definition of $\tilde{N}_{mn}^{xy}$ is similar to N_{mn}^{xy} in the context of short subsequence. With the similar derivation as above, which can be also seen in (20), the metric $\Lambda^{xy}(\tilde{Q}^x, \tilde{Q}^y, \tilde{S}^y)$ can be obtained as follows

$$\begin{aligned} Pr(\tilde{\mathcal{S}}|\mathcal{S}_\tau) = \Lambda^{xy} = \prod_{m=1}^{|Q^x| \times |Q^y|} & \frac{(\tilde{N}_m^{xy})!(N_m^{xy} + |\Sigma^y| - 1)!}{(\tilde{N}_m^{xy} + N_m^{xy} + |\Sigma^y| - 1)!} \\ & \prod_{n=1}^{|\Sigma^y|} \frac{(\tilde{N}_{mn}^{xy} + N_{mn}^{xy})!}{(\tilde{N}_{mn}^{xy})!(N_{mn}^{xy})!} \end{aligned} \tag{3.17}$$

3.7 Transfer entropy as an importance metric

As a pattern based algorithm we use a metric to capture how important is the interaction between two time series at a given instant (for a dynamical system having multiple time series). When inferring about the nature of the observed data, we consider short time sequences which are collected as the system operates. For each pair of variables X and Y in the n-point time window of (x_n, y_n), we calculate the importance metric. The importance metric T^{xy} should have two desirable properties:

1. It should reflect the degree to which the full model (the model learned by using joint state space consideration of the target time series and the source time series which is suspected to have influence on the target time series, or joint patterns in short) yields a better prediction of the target variable than the reduced model (the model learned by using the state space of the target time series only, or atomic patterns in short), as inferred from the given time window.

2. It should easily reflect the importance of the current (joint) pattern with respect to the learned nominal pattern τ^{xy} in the modeling phase.

As a framework that detects anomalies based on dynamics of changes in the influence that one time series exerts on another (in a multivariate time series setting), we believe that property 1 is more important than property 2. Transfer entropy has been used successfully in several applications (9; 10; 11) to estimate this desirable property that we mention in 1. However a tricky issue in evaluating this empirical metric is to come up with accurate estimates of conditional and joint probabilities to find out conditional entropies. A method that would use our Dirichlet priors of joint symbol generation matrix and self symbol generation matrix to improve transfer entropy estimate is left as future work. It is difficult to evaluate the importance of test data based on prior nominal training data, based on transfer entropy alone. It is here that we believe that the metrics Λ and λ would be specially useful in characterizing various stages of the dynamical system. In our experiments we use transfer entropy as an importance metric. However to prevent overestimation from small time windows, we estimate our conditional and joint probabilities from the entire state space transition matrix observed from nominal training data as well as the transition matrix observed from the small time window of test data.

3.8 Transfer entropy is a Granger causal metric

Following the proof from (15), it is easy to show that for the multivariate Gaussian case, the transfer entropy estimator is then just half of the Granger Causality estimator from Y to X. However for any linear finite-order VAR model, we approach the equivalence, albeit only asymptotically.

Hlavackova-Schindler (18) extended the equivalence between transfer entropy and Granger causality to generalized normal variable, which is described as the following lemma for characterizing the main result in this work.

Lemma 3.8.1 *((15)) For generalized normal variables, the Granger causality and transfer entropy are equivalent up to a factor of 2.*

In the symbolic domain, the transfer entropy is calculated based on different symbol sequences associated with different dynamic systems. Therefore, Figure 3.1 shows the STPN frameworks in which the atomic and relational patterns involve the history of observed data. The following lemma states the relationship between transfer entropy in the symbolic domain and Granger causality.

Lemma 3.8.2 *(Theorem 4.2 (18)) For generalized normal variables, the Granger causality and transfer entropy in the symbolic domain are equivalent up to a factor of less than or equal to 2.*

The proof of Lemma 3.8.2 follows from Lemma 3.8.1. It suggests that a finer symbolization can improve the factor, but probably making the transfer entropy computationally intractable. A later discussion for computing transfer entropy in practice is given. With Lemma 3.8.2 in hand, we are ready to state the main theorem in this work.

Theorem 3.8.1 *For any bivariate finite-alphabet stationary ergodic Markov process (I, J), the Granger causality and transfer entropy in the symbolic domain are equivalent.*

(Sketch) For a bivariate finite-alphabet stationary ergodic Markov process (I, J), we first denote by $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ the incoming symbol j_{n+1}, the history of process I, i_n^k, and the history of process of J, j_n^k, given the current time is n. Based on Lemma 3.8.2, and the analysis in (18), we can show that the Granger causality and transfer entropy for the discrete variables in symbolic domain are equivalent to each other.

Such a theorem immediately implies the following corollary for the G-STPN.

Corollary 3.8.1 *G-STPN is a Granger causal framework.*

Remark 3.8.1 *Theorem 3.8.1 implies a significantly important application to time-series inference under the Markov property assumption. For two different dynamic systems or variables, their Granger-causal relationship can be quantified by the transfer entropy. The Corollary 3.8.1 extends the STPN to Granger causal STPN and first time, to our best knowledge, shows rigorously that G-STPN is Granger-causal framework.*

Remark 3.8.2 *To summarize, we describe an extension to STPN, the G-STPN which is shown to have the capability to capture Granger causality across the subsystems, by the properties of the proposed importance metric. Finally we recall that transfer entropy is equivalent to Granger causality in multivariate Gaussian case and asymptotically approaches the original formulation for Granger causality for any linear finite order VAR model.*

As we train an RBM in the later sections, we expect to capture the probability distribution of $\hat{\mathcal{T}}_{X->Y}$ in the form of nominal probability distributions on nominal data. This would in turn capture the true transfer entropy between two variables as a function of time based on theorem 3.1.

CHAPTER 4. AN STPN BASED FRAMEWORK FOR ANOMALY DETECTION AND ROOT CAUSE ANALYSIS

Upon learning STPN or G-STPN from multivariate time series data, we can leverage a combined learning framework based on STPN and Restricted Boltzmann Machine (RBM) introduced in (14; 26) for anomaly detection and root-cause analysis. While details can be found in the original papers, we briefly describe the framework below for completeness.

4.1 Data preprocessing

In this section we cover the main issues which might be encountered from the theory presented in section 3:

- Finding the optimal number of symbols for symbolizing the (continuous) data collected from a Cyber-Physical-System (CPS).
- Finding the optimal length of history for calculation of importance metric
- Finding the optimal length of a time window for online calculation of importance metric.
- Reshaping data to facilitate parallel computing

4.1.1 Transforming time series data from continuous to symbolic domain

In order to calculate the joint state-symbol transition matrix τ and the self state-symbol transition matrix Π, we would require to approximate the continuous time series using a fixed number of symbols. The process of generating multidimensional symbol sequences from a multidimensional continuous time series is known as partitioning the time series. Arbitrary assignment of cotinuous interval to symbols may result in erroneous representation of the original time-series. Thus several well known criteria for creating such partitions exists such as Kolmogorov-Smirnoff distance (27; 28), minimum misclassification cost (29), wavelet based partitioning (30), and false nearest neighbor based (31). In our case we use the Marginal Maximum entropy Partitioning (MMEP). Detailed theoretical discussion about MMEP is provided in (32). Following the maximum entropy criterion, a time series may be partitioned in such a way so as to maximize the overall entropy of the partition. If each interval created by two partitions is denoted by the event J, the entropy of the partition which we seek to maximize is given by:

$$H = -\sum_{j=1}^{J} P_j log P_j \tag{4.1}$$

Where P_j is the probability associated with the event E_j, which is often calculated using frequency counting. Then, according to the MMEP criterion, "to approximately maximize the entropy of a d-dimensional partition for a d-dimensional time series, ($d \geq 2$), maximize the entropy of each one dimensional partition" (32).

4.1.2 Phase space reconstruction and optimal length of history

To find out the components τ and Π of our G-STPN, the state spaces of the interacting subsystems need to be reconstructed from the symbolized multivariate time series data. For such purposes, we would use Takens delay embedding (33) with the suggested parameters by Ragwitz (34) for each component of the multivariate time-series. In the construction of states of a component X, two main parameters come to play, the embedding delay κ and the embedding dimension m which determines how far back do we look into the history of the process. Then each state of X is determined by observing the unique symbol sequence $X_t^{(m)} = (X_t X_{t-\kappa} X_{t-2\kappa} ... X_{t-(m-1)\kappa})$. For an infinitely long time series with infinite precision, all values of the delay parameter κ are equivalent. However, in practical situations, too large a value of κ may force independence in the successive elements of the embedding vector. Too small a κ may again introduce strong correlation between the successive embedded elements. In (35), the authors consider the first zero of the autocorrelation function of X_t as the simplest reasonable estimate of the optimal delay. The autocorrelation function is given by $\rho(k) = \gamma(k)/\gamma(0)$, where $\gamma(k) = Cov(X_t, X_{t+k})$, $k \in \mathbb{Z}$. Following the same method in (35), we perform the nested null hypothesis test

$$H_0 : \rho(k) = 0 \quad versus \quad H_1 : \rho(k) \neq 0 \tag{4.2}$$

The next issue is selection of the embedding dimension m. One popular model free method is the false nearest neighbors (36). In this method we successively calculate how far the m-valued and the (m+1) valued state space vectors are distanced from its k-nearest neighbors. The neighbors are declared to be false if the distance grows too much with increase in dimension and the algorithm increments the value of the embedding dimension m by 1 until no false neighbors are found.

4.1.3 Optimal length of time-window

In the proposed online anomaly detection framework, we collect time windows of streaming data for a fixed length and make our decision over that window. We use overlapping windows in our analysis of root-causes so that we wait till the required number of skip data points have been collected for the next window (also called stride). We note that anomaly in a time series can be caused by one or more anomalous subsequences. Choosing the window size and stride length is important as larger values may fail to capture important local features while lower values may result in too high rate of false alarms due to noise. Most of the time-window based techniques have a user defined window size parameter. Some of the techniques, as described in (37; 38; 39) use the window size equal to the periodicity of the data which is computed using the correlogram of the time series. A correlogram is a graphical representation of the auto correlation functions of the time series which are plotted with different lags. It is assumed that if the time series is periodic then the correlation of the phased lagged version of the time series with its original would produce a maximum value, if the phase lag is equal to its periodicity. The correlogram also shares the same periodicity as the process that generates the original time series (38).

4.1.4 Data reshaping

The streaming time series data is stored as a column vector of words in such a way that ensures re computation of the same values of probability estimators for transfer entropy is avoided. The table consists of a number of rows which is equal to the size of the time window. On encountering new streaming data of the size of stride length, some rows are deleted and the some rows are filled with the new data. Each row stores a unique word in the format $s^y q^x q^y$, where s^y is an unique identifier for a symbol in Y, q^x is a unique identifier for a state in X and q^y is an identifier for a state in Y. Each word is also assigned a unique numerical value in an ascending order based on the number of unique words encountered in total, and is calculated by using the leftmost bit as the least significant bit. The ordering of the whole column is always maintained in an ascending order and new elements are inserted into their respected ordered positions using insertion sort. This arrangement facilitates fast calculation of parameters for the various maximum likelihood probability estimates used in the equation for transfer entropy.

4.2 Extraction of spatio-temporal patterns

We present the combined STPN+RBM framework to capture the nominal robotic operations using multivariate time-series data below. Please see (1) for a detail technical description of this recently proposed framework.

1. Perform maximum entropy partitioning of multivariate time series into a N number of bins and symbolize the time series as decided by the strategy described in earlier;

2. For a pair of variables a and b find the state transition matrix Π^{ab} where Π^{a}_{ij} denotes the probability of transition from state i in state sequence a to symbol j in symbol sequence a;

3. For a pair of variables a and b find the joint state transition matrix τ^{ab} where τ^{ab}_{ijk} denotes the probability of transition from state i in state sequence a and state j in state sequence b to symbol k in symbol sequence b;

4. Consider short sub-sequences of symbols from the training sequence and evaluate T^{ab} $\forall a, b$ for each short subsequence;

5. We assign T^{ab} $\forall a, b$ by calculating the transfer entropy from a to b. The real values obtained are converted to binary patterns by k-nearest neighbor thresholding.

6. Note, we consider a fixed time window W to select the short sub-sequences which is moved over the entire time series. The binary pattern P^{ab} $\forall a, b$ for one short sub-sequence (or time window) becomes an input pattern for the RBM training. We get many such patterns (all considered nominal) from all the time windows defined for the training data. The RBM is then trained to assign a low free energy value (or high probability of occurrence) to these training patterns.

4.3 Data driven system calibration by Restricted Boltzmann Machine (RBM) and subsequent anomaly detection

RBM is an energy based modelling method that takes in an input vector $v = (v_1, v_2, ..., v_D)$, maps it to a set of hidden vectors $h = (h_1, h_2, ...h_F) = Wv + b$, where W is the weight matrix and b is the bias vector. The energy of the configuration (v,h) is defined using the Boltzmann distribution so that the probability of the configuration is:

$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \tag{4.3}$$

W and b are typically obtained by maximizing the likelihood of the training data. RBM can be used to capture multiple operating modes in the system as well as a collection of weak learners forming a strong learner with STPN interpreting the Granger-causality. For a graphical model consisting of n subsystems, all the joint and self patterns together form a binary vector v of length $L = n \times n$. One such binary vector is treated as one training example for the system-wide RBM (with n^2 number of visible units) and many such examples are generated from different short sub-sequences extracted from the overall training sequence. Then, the RBM is trained by maximizing the maximum likelihood of the data, thus calibrating the anomaly detection system for nominal data.

During training, weights and biases are obtained such that the training data has low energy. During inference then, an anomalous pattern should manifest itself as a low probability (high energy) configuration. The energy function for an RBM is defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h}$$

where $\mathbf{W}$ are the weights of the hidden units, $\mathbf{b}$ and $\mathbf{c}$ are the biases of the visible units and hidden units respectively.

With the weights and biases of RBM, free energy can be computed. Free energy is defined as the energy that a single visible layer pattern would need to have in order to have the same probability as all of the configurations that contain $\mathbf{v}$ (40), which has the following expression:

$$F(v) = -\sum_i v_i a_i - \sum_j \ln(1 + e^{b_j + \sum_i v_i w_{ij}})$$

The free energy in nominal conditions is noted as $\tilde{F}$. In cases where there are multiple input vectors with more than one nominal mode, free energy in the nominal states can be averaged or used in conjunction with other metrics. In anomalous conditions, a failed pattern will shift the energy from a lower state to a higher state. If the entire pattern obtained during test consists of nominal and anomalous patterns, $\mathbf{v}^{nom}$ and $\mathbf{v}^{ano}$, the combined free energy can be obtained as:

$$\begin{aligned} F^s(v) = &- \sum_g v_g a_g - \sum_j \ln(1 + e^{b_j + \sum_g v_g w_{gj}}) \\ &- \sum_h v_h a_h - \sum_j \ln(1 + e^{b_j + \sum_h v_h w_{hj}}), \\ &\{v_g\} \in v^{nom}, \{v_h\} \in v^{ano} \end{aligned}$$

An anomalous mode of operation is immediately detected when the value of the free energy departs significantly from that obtained in case of nominal operation, denoting that a low probability event has occurred. Although noting the change in the value of free energy in real time may conclusively suggest anomalous operation, it does not tell anything about the cause of the failure, thus a root cause analysis technique tries to find out the most probable subset of v_h that explains the deviation in free energy.

4.4 Locating anomalous system operation via root-cause analysis

In the combined STPN+RBM framework described above, anomaly manifests itself as a low probability (high energy) event. Therefore, the idea for root-cause analysis is to find potential pattern(s) that, if changed, can transition the system from a high to a low energy state. With the detected anomalous patterns, node inference (locating anomalous manipulator joint) is to find a subset that can interpret all the found failed patterns.

In a high level conceptual manner, the way Root Cause Analysis (RCA) works is by introducing deliberate changes in the patterns (time indexed series of importance metrics) collected during test in real time ($\hat{P}^{XY}$) (Note, however that the patterns need not necessarily be binary). The problem is thus of a sequential search for finding those deliberate changes that transforms the anomalous patterns to nominal patterns in the closest sense possible. For the binary case, it is easy to see that such a search progressively flips short sub-patterns from 0 to 1 and vice versa, thus making it a tractable process unlike patterns constituted of real numbers. More formally, let us suppose that an inference metric T^{XY}_{ano} changes to $T^{XY}_{ano}{}'$ due to an artificial perturbation in the pattern $X \to Y$. We now consider a subset of patterns, for which we have the set of inference metrics $\{T_{ano}^{XY}\} \subset T_{ano}$. Let a perturbation in this subset changes the overall set of metrics to T'_{ano}. Therefore, we have the following:

$$T'_{ano} = \begin{cases} T_{ano}^{XY} \;\; \text{if} \;\; T_{ano}^{XY} \notin \{T_{ano}^{XY}\} \\ \\ T_{ano}^{XY\prime} \;\; \text{if} \;\; T_{ano}^{XY} \in \{T_{ano}^{XY}\} \end{cases} \tag{4.4}$$

The remaining task is to find the optimal subset of patterns such that the following condition is satisfied :

$$\{T_{ano}^{XY}\}^{\star} = \min_{\{T_{ano}^{XY}\}} \mathscr{D}\Big(T'_{ano}, T_{nom}\Big), \tag{4.5}$$

Here $\mathscr{D}$ is the Kullback-Liebler distance metric (KLD) (42). KLD is used because it may be more robust as it takes multiple sub-sequences into account because a persistent anomaly across the subsequences will cause a significant impact on KLD.

4.5 Sequential state switching (S^3)

In the STPN+RBM framework described above, anomaly manifests itself as a low probability (high energy) event. Therefore, the idea for S^3 is to find potential pattern(s) that, if changed, can transition the system from a high to a low energy state. The probabilities of existence of joint patterns are discovered by the STPN, and an anomaly will influence the causality of specific patterns. Hence, by switching/flipping binary patterns P, its contribution on the energy states of the system can be identified and a large contribution may indicate the root-cause of an anomaly.

It should be noted that free energy F is used in Algorithm 1 to be applied as the distance metric of the Eq. 4.5, and it can be used along with other metrics such as KLD. Using the more robust KLD alongside with free energy is particularly useful when the distribution of free energy is obtained with multiple sub-sequences.

Algorithm 1 Root-cause analysis with sequential state switching (S^3) method

```
procedure STPN+RBM MODELING    ▷ Algorithm discussed in section 4.2
    Use the maximum likelihood estimates to compute probability of joint state-symbol transitions
    Use the probabilities to compute importance metrics $T$
    $P \leftarrow$ array of 0 of length $= length(T)$    ▷ Stores the digitized patterns $P$ of the G-STPN
    while i $\leq$ length($T$):
        $P[i]$=knn($T$) ▷ knn is a procedure that takes a real number as input and output 0 if its less than a determined threshold based on elements in $T$, else 1
    end while
    Use extracted patterns of G-STPN ($P$) to train RBM that assigns high probability(low energy) to nominal data.
end procedure
procedure ANOMALY DETECTION    ▷ Algorithm discussed in section 4.3
    Online anomaly detection via observing the KL divergence of the free energy of RBM in testing phase from that in training phase.
end procedure
procedure ROOT-CAUSE ANALYSIS
    if $Anomaly = True$ then
        $E_0^c \leftarrow E^s(\mathbf{v})$ ▷ $E^c$ is the current free energy with input vector $\mathbf{v} = \mathbf{v}^{nom} \cup \mathbf{v}^{ano}$.
        $\mathbf{v}_p \leftarrow \{v : E^s(\mathbf{v}) < E^c\}$
        $\mathbf{v}^{ano} = \emptyset$
        $\mathbf{v}^{nom} = \mathbf{v}$
        while $\mathbf{v}_p \neq \emptyset \vee \{v : E^s(\mathbf{v}^{\star,ano}, \mathbf{v}^{nom}) < E^c\} = \emptyset$ do
            $E^c \leftarrow \min(E^s(\mathbf{v}^{\star,ano} \cup v_i^\star, \mathbf{v}^{nom}))$, $v_i \in \mathbf{v}_p$, $v_i^\star = 1 - v_i$, $\mathbf{v}^{\star,ano} = 1 - \mathbf{v}^{ano}$
            $\mathbf{v}^{ano} \leftarrow \mathbf{v}^{ano} \cup \{v_i : E^s(\mathbf{v}^{\star,ano} \cup v_i^\star, \mathbf{v}^{nom}) = E^c\}$
            $\mathbf{v}^{nom} \leftarrow \mathbf{v}^{nom} \setminus \mathbf{v}^{ano}$
            $\mathbf{v}_p \leftarrow \mathbf{v}_p \setminus \{v_i : E^s(\mathbf{v}^{\star,ano} \cup v_i^\star, \mathbf{v}^{nom}) = E^c\}$
        end while
    end if
    A bijective function: $g : \mathbf{P} \rightarrow \mathbf{v}$
    $\mathbf{P}^{ano} = g^{-1}(\mathbf{v}^{ano})$
    return $\mathbf{P}^{ano}$
end procedure
```

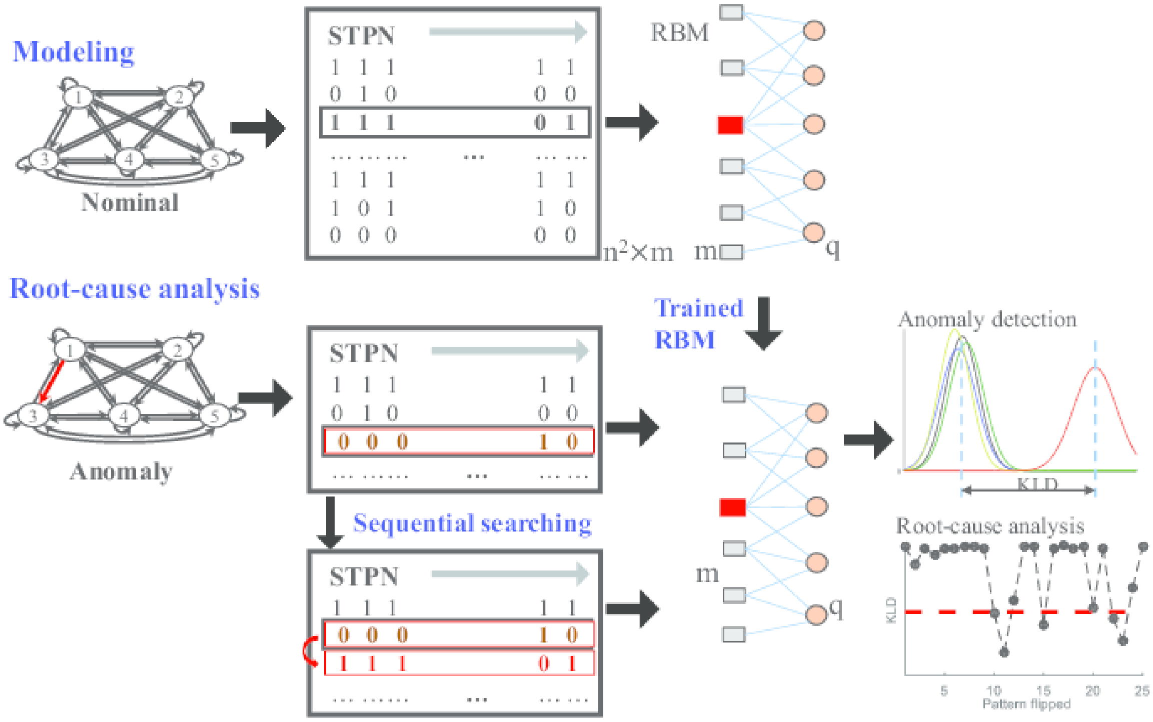


Figure 4.1 A two-state case of anomalous condition with two sub-systems. The state transitions between the subsystems a and b are first defined in the nominal condition (shown in the top panel) assuming the depth $D = 1$ and the time lag $p = 1$. Then, in the anomalous condition, changes occur from the state q_1^a to the state q_2^a in the subsystem a and we assume that the changes only exist in the transitions $q_1^a \to q_1^b$ (i.e., they change to $q_2^a \to q_1^b$ due to the anomaly). The Hamming distance between the sequence for the subsystem a in the anomalous condition and that in the nominal condition is η^a.

CHAPTER 5. A PRACTICAL CASE OF ANOMALY DETECTION

We now move on to an experimental case study to show that STPN or G-STPN can capture the dynamics of a system using multivariate time series originating from the system, such that anomalous operations of the system can be discriminated from nominal operation and the source of the anomaly can also be isolated. In literature, this is known as anomaly detection and root cause analysis which is crucial for safety-critical and attack resilient systems.

5.1 Motivation and existing literature

Majority of the manufacturing industries have seen a tremendous rise of industrial robot usage in recent years. With the advent of service robots such as Baxter, Sawyer and Kuka (43; 44; 45; 46), there has been an increased focus in deploying these user friendly robots in industrial environments alongside their human counterparts. In this age of Internet of Things (IOT), most industrial robots need to be connected to the internet (intranet) to enable remote communication with the operator and/or other robots. This opens a window for hackers to get access to and manipulate these robots with malicious intent (47). Depending on the final goal, the hacker may use the large quantity of data generated and processed by the robot's sensors for espionage as well as to simply sabotage the operation of the industry by altering the operation of the robot and/or disabling actuators (48).

There has been numerous recent research in anomaly detection and characterization for robotic systems. In (49), sequential image data is extracted from the robot system using monocular camera and mapped to a lower dimensional feature space using convolutional neural networks. Anomalies are detected by comparing the actual data with the predicted behavior using the learnt features. In (50), multi-modal observations are collected from non-anomalous performances and used to train a Hidden-Markov Model (HMM). A combination of self-organizing maps (SOM) with spatial context mapping for the spatial domain and generative probabilistic graphical models for the temporal domain is used in (51) to detect anomalies. In (52), the authors propose to generate adaptive thresholds using locally linear models (LLM) and Model Error Modeling (MEM) techniques for fault detection on two wheeled mobile robot. Using Variational autoencoders (VAE) with Recurrent Neural Networks (RNNs) as encoders and decoders as a generative time series model, the authors in (53) showed that anomalous cases can be detected successfully. Other recent works using hybrid timed automata (54; 55), k nearest neighbors (56), sensor redundancy (57), and various robot introspective frameworks (58; 59) have also been done for anomaly detection in robotics systems.

While physical failures occurring within a robot can be easily observed and monitored, intelligent hacks meant to reduce overall productivity can be very hard to detect. In this work, we discuss three main ways in which a robotic operation can be sabotaged in an intelligent manner and ways to detect such anomalies occurring in the system. We propose to use the concepts of the spatiotemporal pattern networks (STPN) (1) to learn the temporal characteristics of individual variables involved in the robotic system as well as Granger-causal relationships (60; 15) among the different variables.

5.2 Data and testbed

In this work we have used time series data collected from the one-handed manipulator Sawyer for simulating and detecting anomalies. A schematic of the Sawyer robot showing the degrees of freedom and joint assignment is shown in figure 5.1. This will be necessary for indicating the anomalies. In this experiment, the data collected is spatially correlated multivariate time-series.

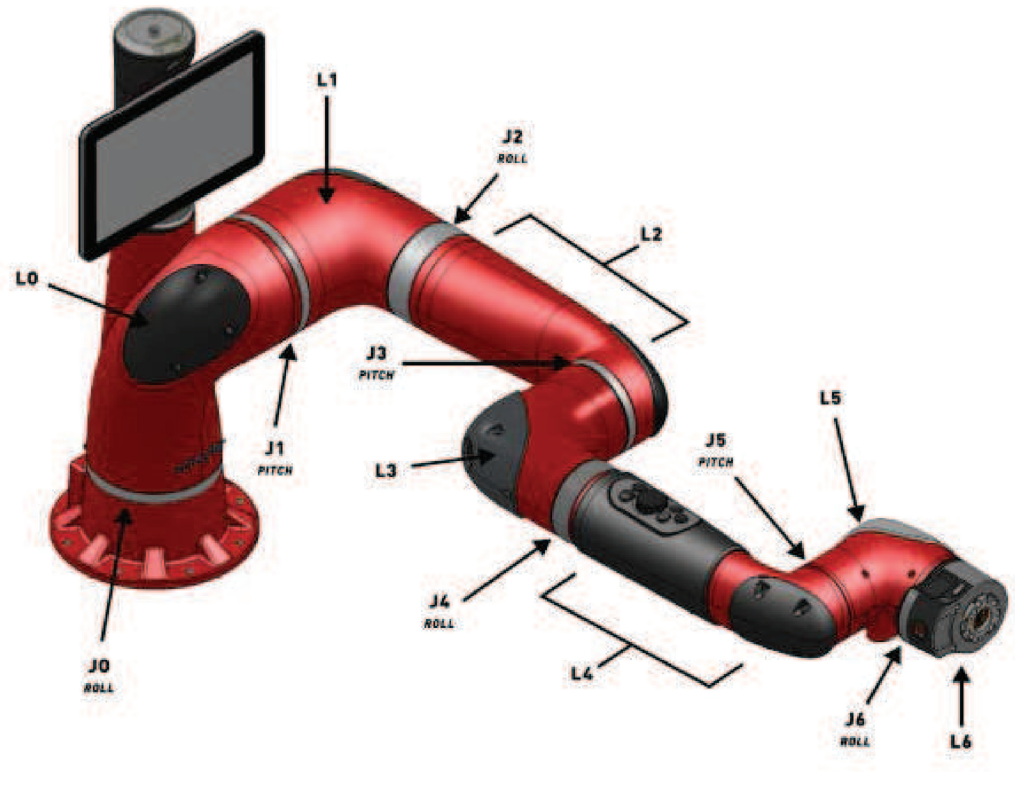


Figure 5.1 Description of the joints

The data collected has 27 controlled variables denoting the angular position, velocity and acceleration for each of the seven joints (joints J0 through J6 in the figure 5.1), the head pan and the linear position, velocity and acceleration of the gripper. The remaining 19 observed variables denote the five 3-dimensional variables (i.e., position, linear twist, angular twist, wrench force, and wrench torque) for x,y and z dimensions and one 4-dimensional variable (i.e., orientation) of the end effector. By the experimental setup, there is a causal relationship between controlled and observed variables. In the different ways as mentioned in section 6 we change controlled variables, simulating a cyber-attack, and try to detect the change using only observed variables.

5.3 Comparing STPN and G-STPN

Before analyzing the data for anomaly detection and root-cause analysis, we numerically compare the information captured by STPN and G-STPN. Note that for both STPN and G-STPN, the importance metrics (mutual information for relational patterns and transfer entropy for joint patterns respectively) are extremely crucial. The importance metric is central to the STPN+RBM framework for anomaly detection and root cause analysis. In most of the earlier work using STPN, joint patterns were not considered while calculating importance metrics. Earlier we also mention in remark 3.2.1 that as a consequence it may be difficult to "condition out" past histories. In table 1, we compare the information values extracted from relational patterns as in the STPN framework used in (13) and the information values extracted from joint patterns in our G-STPN framework. We note that as the length of history increases, more the gap becomes more apparent with the G-STPN importance metric increasing in comparison to the other metric and gradually plateauing at length of history equal to 3, from whereon it eventually reduces in value. All the while the value of Relational Pattern(RP) in STPN stabilizes at length of history equal to 1, with minor fluctuations at increasing depths.

It seems that an the increase in value of information is achieved at a forbiddingly high computational complexity for G-STPN. Ideally it involves searching through the joint (three dimensional) product space of states and symbols of the two sub-systems in consideration. However we note that:

- Structured data (in most real cases) after symbolization is supposed to contain only a fixed number of observable transitions in the corresponding Markov process, thus although the search space is of the dimension $|S^Y| \times |Q^X| \times |Q^Y|$, the actual number of unique observed transitions $q^x \times q^y \longrightarrow s^y$ would be considerably lower.

- The calculation of transfer entropy can be significantly accelerated by parallel computation, following a pre-indexing technique as described in section 4.2. Table 1 also gives the corresponding time in seconds used to calculate importance metrics for G-STPN.

We now describe the experiment used to investigate the difference between the two methods empirically. Considering the dynamics of the robot, the base joint (joint 0), as shown in figure 5.1, highly influences the movement of other joints in the serially connected links. Thus, a metric that is designed to capture causality would be appropriately used in determining importance metrics from the joint 0 to a close link (joint 3) and to the last link (joint 6). We gradually vary the depth of the Markov Machines from 1 to 3. We observe that the difference between transfer entropy and the mutual information gradually increases with increase in depth as shown in the Table 5.1.

As discussed before, finding the optimal length of history, the number of symbols for symbolization of continuous data, and the length of collected data that is good enough for unbiased estimation of transfer entropy are three main continuing challenges in the field. We address these issues in section 4. In the sequel, we present the detailed experimental setup and the anomaly detection and root-cause results.

Table 5.1 Transfer entropy (TE) for Joint Patterns(JP) and Mutual Information (MI) for Relational Patterns(RP)

Variable Pair	Depth	TE for JP	MI for RP	Time taken for TE
		value (bits)	value (bits)	value (seconds)
joint 0 to joint 3 positions	1	0.907	0.920	7.0
	2	1.396	0.974	15.0
	3	1.481	0.828	20.6
	4	1.35	0.801	38.0
joint 0 to joint 6 positions	1	0.827	0.792	6.5
	2	1.317	0.878	14.3
	3	1.403	0.794	22.0
	4	1.27	0.750	35.3

CHAPTER 6. EXPERIMENTS, RESULTS AND DISCUSSION

6.1 Attack injection

All the attack types are visualized with time series of image frames in figure 6.1.

Type 1: Controller hack: For controller hack we simply change the controlled value of joint 0, 2, 3, and 4, four arbitrarily selected joints. The values of joints 0 and 3 is multiplied by a time-dependent error factor e_i which has the following form

$$e_i = \begin{cases} \frac{|c-i|}{c}, & \text{if } i \leq c \\ \frac{|i|}{c}, & \text{otherwise } i{>}c. \end{cases}$$

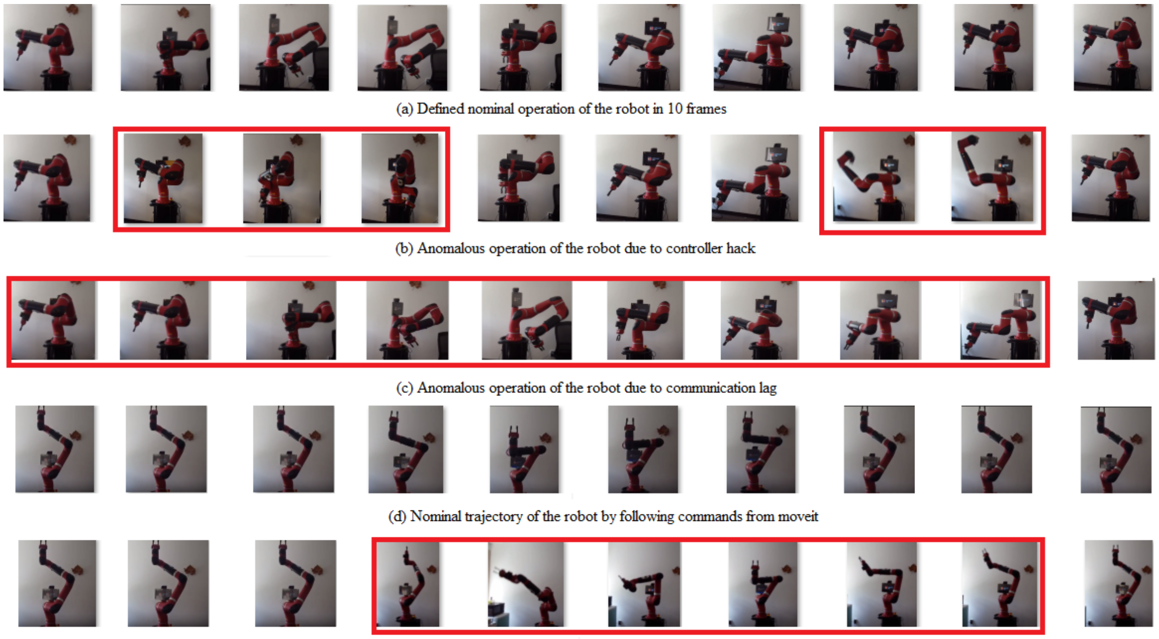
where, c is a positive constant defined specifically in the experiments and i is the time index of first c points in dataset Q. Then the values of joints 2 and 4 is multiplied by the same time-dependent error factor e_i with a time shift.

$$e_i = \begin{cases} \frac{|d+c-i|}{c}, & \text{if } i \leq d + \frac{c}{2} \\ \frac{|i-d|}{c}, & \text{otherwise } i{>}d + \frac{c}{2}. \end{cases}$$

where, c and d are positive constants defined specifically in the experiments and i is the time index in dataset Q between indices d and $d + c$.

Type 2: Communication delay: For simulating communication delay, we set the first c controlled values of joint 3 equal to the value of the first data point (i.e., a zero order hold) and then after the c^{th} data point every value for joint 3 is defined by the following

$$Q(i,3) = \begin{cases} Q(1,3), & \text{if } i \leq c \\ Q(i-c,3) & \text{otherwise } i{>}c. \end{cases}$$



(a) Defined nominal operation of the robot in 10 frames

(b) Anomalous operation of the robot due to controller hack

(c) Anomalous operation of the robot due to communication lag

(d) Nominal trajectory of the robot by following commands from moveit

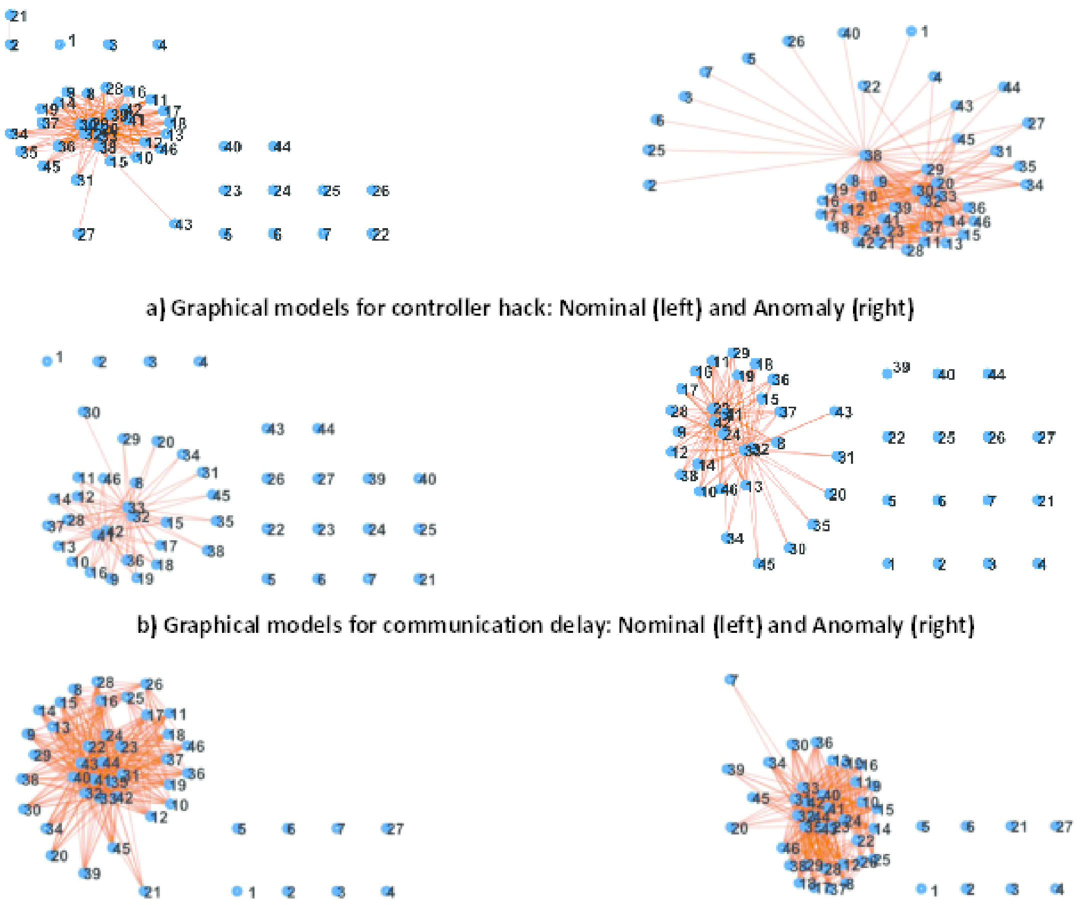
(e) Anomalous trajectory of the robot following anomalous commands from moveit because of imaginary obstacle (pictures taken once in two seconds)

Figure 6.1 Time frames showing the task execution in nominal and anomalous conditions

Type 3: Trajectory manipulation: For simulating intelligent attacks of type (3) we use the MoveIt! inverse kinematics library with visualization in Rviz. The library is typically used to solve the inverse kinematics problem of the desired minimum path length trajectory for a given starting and ending end effector position and orientation. Nominal training data is obtained by collecting C and J for a given starting and ending position and orientation of end effector. Anomaly is introduced by placing an imaginary obstacle at a certain point in the shortest path trajectory forcing the inverse kinematics library to calculate a longer path for the same task.

6.2 STPN+RBM models

The method used in this work is predominantly based on STPN. Thus, in this section we visualize and discuss the probabilistic graphical model representations of our robotic system under the three different anomaly cases. The figure 6.2 below shows the graphical models derived from the underlying raw measurement data. It is the penultimate step in our anomaly detection algorithm after the atomic and relational patterns have been thresholded to a value of either 0 or 1 using K nearest neighbors method and is ready to be fed to the RBM as a series of training data that covers the entire length of the observed nominal data. The graphical representation of the system has been simplified only for visualization purpose. We consider an undirected link between two nodes/variables a and b if there is a significant relationship between the variables from both directions as evidenced by the metric T^{ab}. Note that the metric values are averaged over all time windows during both nominal and anomalous conditions and we only consider the links relevant for the injected anomalies. In figure 6.2, we can clearly see the changes in overall graph connectivity due to the injected anomalies. Implicitly, such results demonstrate that the proposed STPN framework can capture the anomalies of different types for the robotic system.



a) Graphical models for controller hack: Nominal (left) and Anomaly (right)

b) Graphical models for communication delay: Nominal (left) and Anomaly (right)

c) Graphical models for trajectory manipulation: Nominal (left) and Anomaly (right)

Figure 6.2 Derived graphical models with 3 types of anomalies: (a) controller hack; (b) communication delay; (c) trajectory manipulation

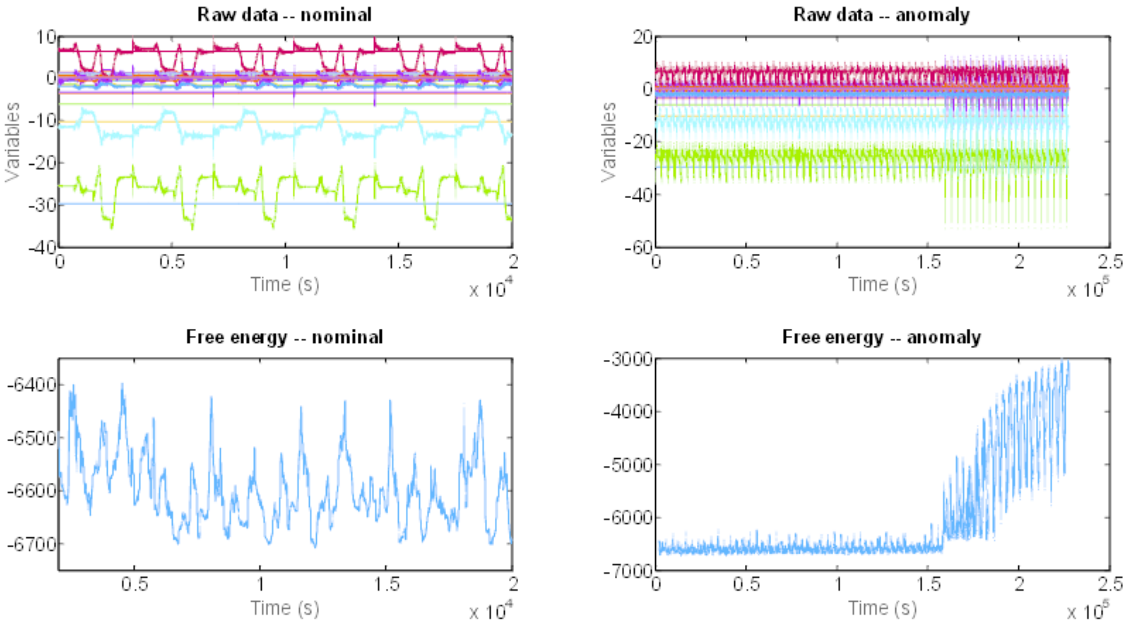


Figure 6.3 Raw Data and free energy of RBM under nominal and anomalous conditions for the controller hack

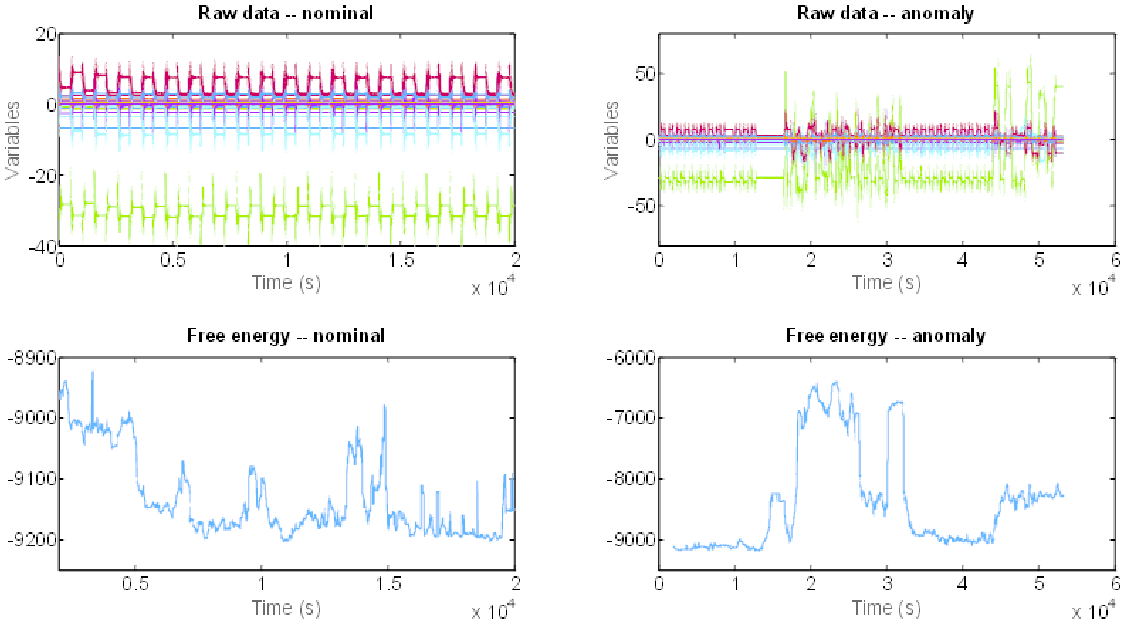


Figure 6.4 Raw Data and free energy of RBM under nominal and anomalous conditions for the trajectory manipulation

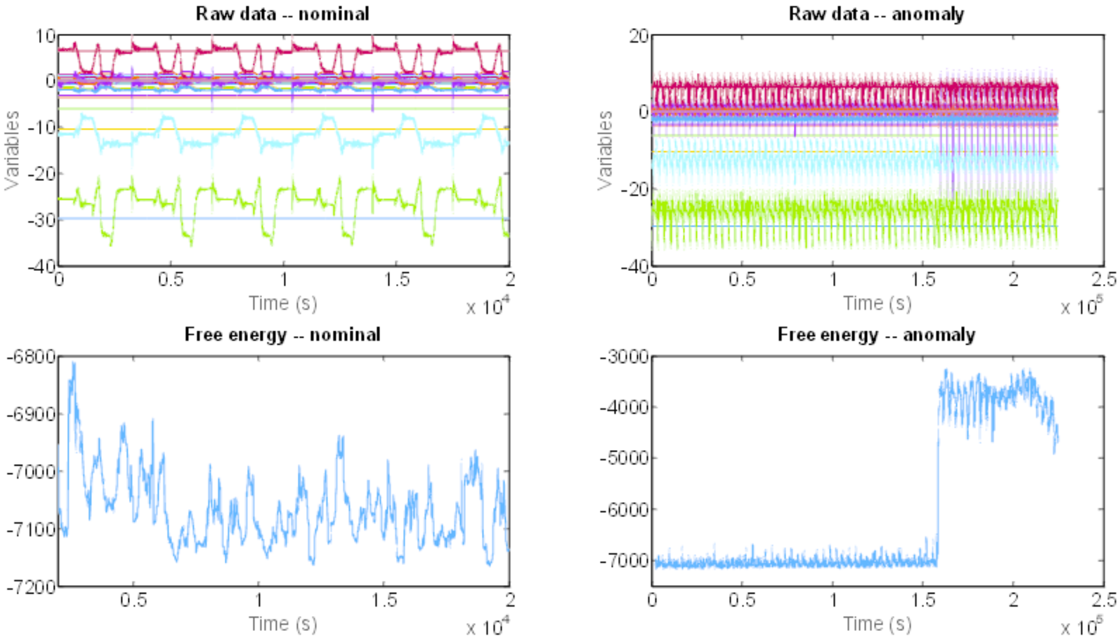


Figure 6.5 Raw Data and free energy of RBM under nominal and anomalous conditions for the communication delay

6.3 Spatio-temporal anomaly detection

The root cause anomaly detection algorithm proposed is able to detect causes of system anomaly on both spatial and temporal scales providing explanations of failure at each individual subsystem as well as the explanations for failure because of anomalous interactions occurring at certain time points of operation between subsystems. The results essentially present us with a 3 dimensional graph with nodes, time points, and anomaly scores as x, y and z axis respectively. To simplify the interpretation, we take the average of all the anomaly scores along time and along nodes to produce two explanations:

1. Time averaged node anomaly score presented in figure 6.7 for both G-STPN and STPN frameworks

2. Node averaged time anomaly score presented in figure 6.9 for both G-STPN and STPN frameworks

The time period of operation of the robot is roughly 35 seconds. This work is repeatedly performed and recorded as nominal data for around 12 minutes. The frequency of data collection is 100 datapoints per second. The Window size for extracting importance metrics is 2000 points(=20s), the window stride length is 10(=0.1s). During inference in RCA, a collection of 50 windows are considered together(=5s resolution for anomaly detection). We call each of these collection of 50 windows an instant.

For **communication delay**, Joint 4 was programmed to have the time lag anomaly and the time averaged node score (G-STPN) across the nodes shows that node 4 is the most anomalous. In the plot of node averaged time scores (G-STPN-right) peaks are detected at roughly 7 instant intervals (= 35s intervals), the time period of one cycle of operation.

For **controller hack**, in the time averaged node score plot (G-STPN), joint 1,3,4,5 controllers had been programmed for anomaly, for the mid line separating lowest and highest scores, position anomalies of 1,3 and 5 are detected, velocity anomaly of 4 is detected, and acceleration anomaly of 1,3,4 are detected. In the node averaged time score plot (G-STPN) plateaus of about 1 instant(=5s) long are observed (in real anomaly the most observable deviation is also 5s long). As usual, the time gap between the distinct features are around 7 instants (=35 s). It is evident by comparison, that for these considered cases, the results offered by G-STPN is more stable and accurate over the results offered by STPN.

6.4 Performance analysis

Figures 6.3 - 6.5 show the raw data with free energy outputs generated by the trained RBM. For the raw data, we combine the time-series of every variable and show them within one plot. For all of three cases, results show that anomaly patterns in the raw data can be immediately captured by the trained RBM by comparing the nominal and anomalous free energy. It can be observed that the case of communication delay has the most significant variation in free energy for the anomalous condition. The free energy value increases immediately from around -7000 to around -4000. Moreover, compared to the controller hack and

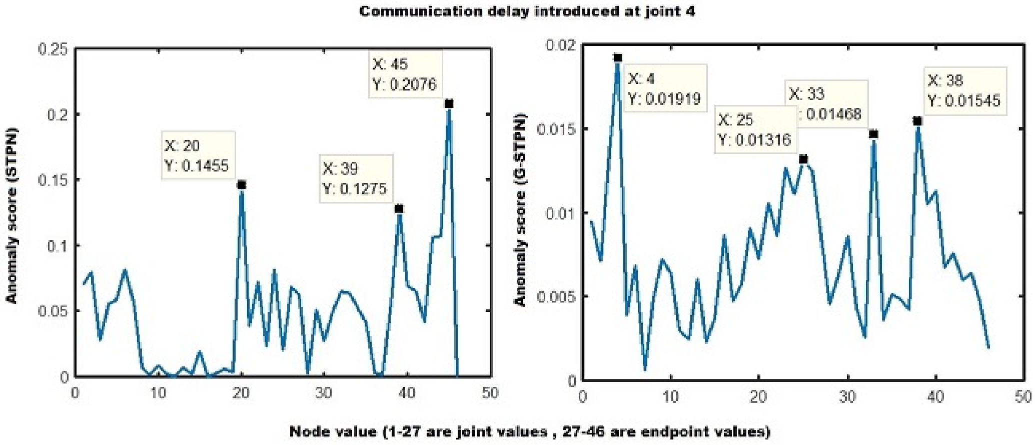


Figure 6.6 Experiment was performed on the robot by repeating cycles of operation for communication delay. We averaged the anomaly scores for each nodes at each time instant to look at the average individual node anomaly score throughout the experiment.

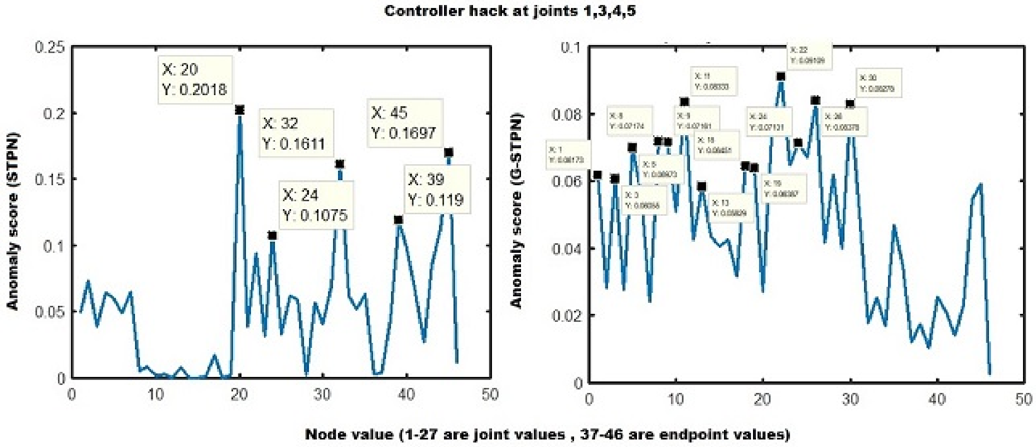


Figure 6.7 Experiment was performed on the robot by repeating cycles of operation for controller hack. We averaged the anomaly scores for each nodes at each time instant to look at the average individual node anomaly score throughout the experiment. G-STPN is able to capture the anomalous angular position, velocity and acceleration for joint 3 in both communication delay and controller hack

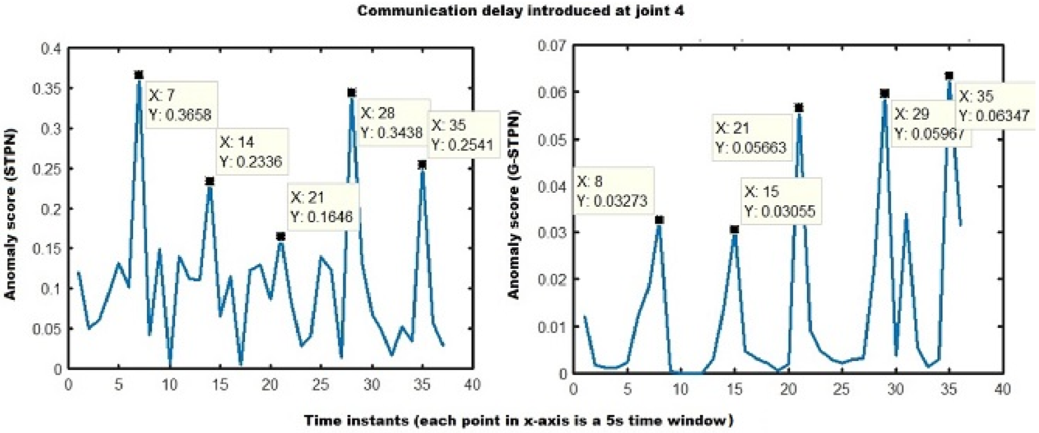


Figure 6.8 Experiment was performed on the robot by repeating cycles of operation for communication delay. We averaged the anomaly scores for each nodes at each time instant to look at the average individual anomaly score at each time instant throughout the experiment across all the nodes. Notice that G-STPN using transfer entropy captures a better signature than STPN

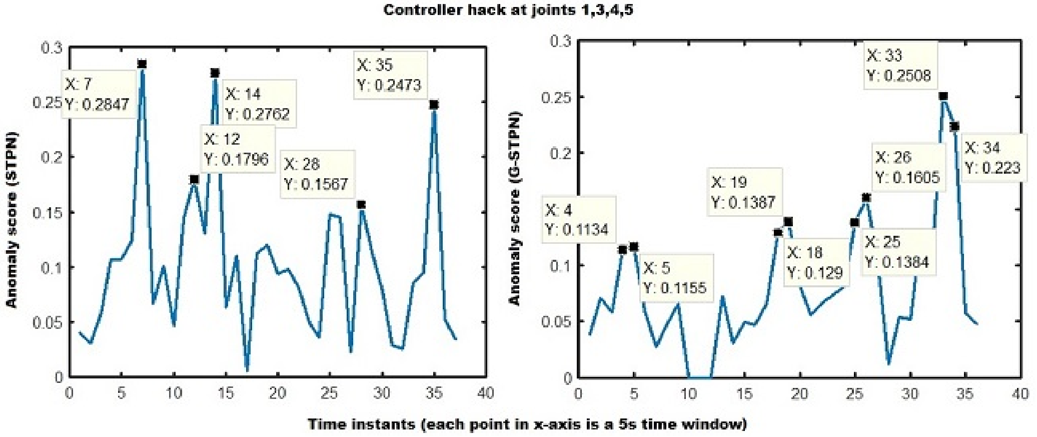


Figure 6.9 Experiment was performed on the robot by repeating cycles of operation for controller hack. We averaged the anomaly scores for each nodes at each time instant to look at the average individual anomaly score at each time instant throughout the experiment across all the nodes. Notice that G-STPN using transfer entropy captures a better signature than STPN

communication delay, the trajectory manipulation shows less significant variation in free energy, which may be attributed to the uncertain joints as most of joints are involved in the trajectories. We now discuss the root-cause analysis results using our proposed scheme. Figure 6.10 shows the root causes corresponding to controller hack and communication delay. In the case of controller hack, anomalies are injected to joints 0, 2, 3, and 4. Along with those, joint 5 is also considered as anomaly because its directly connected to the 2, 3, 4 chains. By using the proposed approach, except joint 5, the rest of four joints can be isolated correctly. As joint 5 is next to joint 4 the isolation of joint 4 may negatively affect the isolation of joint 5. For the communication delay, it can be observed that the anomaly is injected to joint 3. Although eventually using the proposed RCA method enables us to isolate three joints, i.e., joints 0, 3, and 4, where anomalies are detected, joint 3 can be correctly detected to help operators locate the attacks. Joint 4 is also isolated in this case as it is close to joint 3 while for joint 0, the reason may be attributed to the robot's dynamics which is not analyzed in detail in this work. For the case of trajectory manipulation, we can know that when a block is placed in the path from starting point to end, joints can be observed to move for avoiding the block. However, it is difficult to determine the ground truth in terms of joints as the trajectories involve most of the joints. Also, based on the proposed algorithm, results show that most of joints are involved so the isolation of joints are not provided for the root-cause analysis in the trajectory manipulation case.

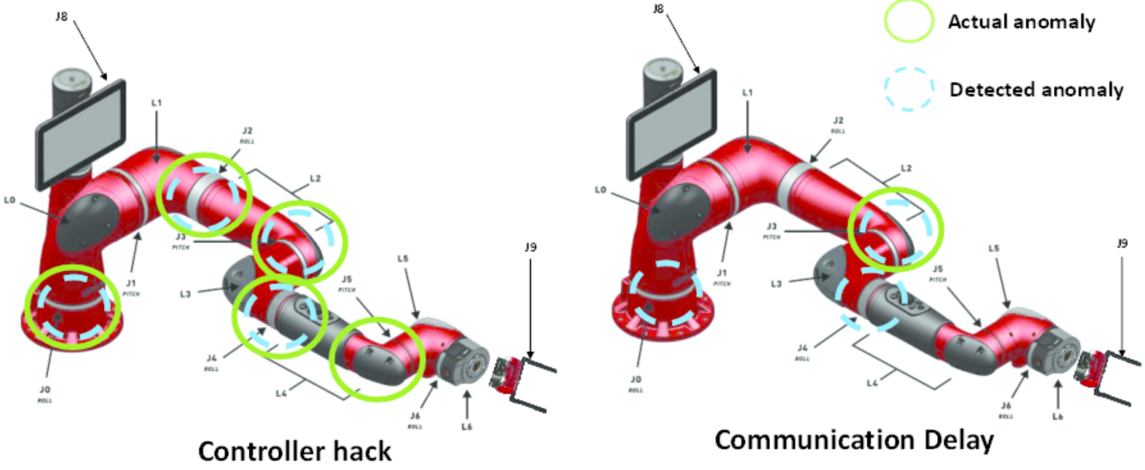


Figure 6.10 Actual anomaly and detected anomaly associated with the defined joints for the controller hack and communication delay

CHAPTER 7. CONCLUSION

This work explored the notion of Granger causality in the context of a recently proposed spatiotemporal graphical modeling technique called, STPN. We show tha Granger causality can be captured by modifying the STPN formulation slightly which leads to a variant called the G-STPN. Upon learning STPN or G-STPN, a combined learning framework involving restricted Boltzmann machines (RBM) can be used to perform anomaly detection and root-cause analysis in complex dynamical systems. We demonstrate the efficacy of such a decision framework using a real experimental case study involving cyber-physical attacks on an industrial robotic platform. We also empirically observe that in certain cases, information content of the STPN patterns can approximate that of G-STPN patterns with significantly lower computational expense. However, a significantly more rigorous study is required to understand this approximation correctly which will be the most important future work. Few other future research directions will include: 1) identifying optimal depth/memory to be considered in STPN or G-STPN; 2) setting up a quantification framework for evaluating the root-cause isolation step; 2) understanding temporal characteristics of attack propagation through a dynamical system; 3) development of on-line attack mitigation strategies.

BIBLIOGRAPHY

[1] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised spatiotemporal graphical modeling approach to anomaly detection in distributed cps," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*, p. 1, IEEE Press, 2016.

[2] C. Liu, A. Akintayo, Z. Jiang, G. P. Henze, and S. Sarkar, "Multivariate exploration of non-intrusive load monitoring via spatiotemporal pattern network," *Applied Energy*, vol. 211, pp. 1106–1122, 2018.

[3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[4] Z. Jiang and S. Sarkar, "Understanding wind turbine interactions using spatiotemporal pattern network," in *ASME 2015 Dynamic Systems and Control Conference*, pp. V001T05A001–V001T05A001, American Society of Mechanical Engineers, 2015.

[5] M. Darianian and M. P. Michael, "Smart home mobile rfid-based internet-of-things systems and services," in *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on*, pp. 116–120, IEEE, 2008.

[6] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual reviews in control*, vol. 36, no. 2, pp. 220–234, 2012.

[7] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark

tennessee eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[8] C. W. Granger, "Causality, cointegration, and control," *Journal of Economic Dynamics and Control*, vol. 12, no. 2-3, pp. 551–559, 1988.

[9] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, "Transfer entropya model-free measure of effective connectivity for the neurosciences," *Journal of computational neuroscience*, vol. 30, no. 1, pp. 45–67, 2011.

[10] T. Dimpfl and F. J. Peter, "Using transfer entropy to measure information flows between financial markets," *Studies in Nonlinear Dynamics and Econometrics*, vol. 17, no. 1, pp. 85–102, 2013.

[11] G. Ver Steeg and A. Galstyan, "Information transfer in social media," in *Proceedings of the 21st international conference on World Wide Web*, pp. 509–518, ACM, 2012.

[12] S. Gupta and A. Ray, "Symbolic dynamic filtering for data-driven pattern recognition," *Pattern recognition: theory and application*, pp. 17–71, 2007.

[13] Z. Jiang, C. Liu, A. Akintayo, G. P. Henze, and S. Sarkar, "Energy prediction using spatiotemporal pattern networks," *Applied Energy*, vol. 206, pp. 1022–1039, 2017.

[14] C. Liu, K. G. Lore, and S. Sarkar, "Data-driven root-cause analysis for distributed system anomalies," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pp. 5745–5750, IEEE, 2017.

[15] L. Barnett, A. B. Barrett, and A. K. Seth, "Granger causality and transfer entropy are equivalent for gaussian variables," *Physical review letters*, vol. 103, no. 23, p. 238701, 2009.

[16] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya, "Causality detection based on information-theoretic approaches in time series analysis," *Physics Reports*, vol. 441, no. 1, pp. 1–46, 2007.

[17] M. Staniek and K. Lehnertz, "Symbolic transfer entropy," *Physical Review Letters*, vol. 100, no. 15, p. 158101, 2008.

[18] K. Schindlerova, "Equivalence of granger causality and transfer entropy: A generalization.," 2011.

[19] S. L. Bressler and A. K. Seth, "Wiener–granger causality: a well established methodology," *Neuroimage*, vol. 58, no. 2, pp. 323–329, 2011.

[20] S. Sarkar, K. Mukherjee, S. Sarkar, and A. Ray, "Symbolic dynamic analysis of transient time series for fault detection in gas turbine engines," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 1, p. 014506, 2013.

[21] L. Barnett and T. Bossomaier, "Transfer entropy as a log-likelihood ratio," *Physical review letters*, vol. 109, no. 13, p. 138105, 2012.

[22] T. A. Severini, *Likelihood methods in statistics.* Oxford University Press, 2000.

[23] P. Billingsley, "Ergodic theory and information," 1965.

[24] A. Wald, "Tests of statistical hypotheses concerning several parameters when the number of observations is large," *Transactions of the American Mathematical society*, vol. 54, no. 3, pp. 426–482, 1943.

[25] J. L. Doob and J. L. Doob, *Stochastic processes*, vol. 7. Wiley New York, 1953.

[26] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised anomaly detection approach using energy-based spatiotemporal graphical modeling," *Cyber-Physical Systems*, vol. 3, no. 1-4, pp. 66–102, 2017.

[27] L. Gordon and R. A. Olshen, “Asymptotically efficient solutions to the classification problem,” *The Annals of Statistics*, pp. 515–533, 1978.

[28] J. H. Friedman, “A recursive partitioning decision rule for nonparametric classification,” *IEEE Transactions on Computers*, no. 4, pp. 404–408, 1977.

[29] E. G. Henrichon and K.-S. Fu, “A nonparametric partitioning procedure for pattern classification,” *IEEE Transactions on Computers*, vol. 100, no. 7, pp. 614–624, 1969.

[30] V. Rajagopalan and A. Ray, “Symbolic time series analysis via wavelet-based partitioning,” *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.

[31] M. B. Kennel and M. Buhl, “Estimating good discrete partitions from observed data: Symbolic false nearest neighbors,” *Physical Review Letters*, vol. 91, no. 8, p. 084102, 2003.

[32] T. Chau, “Marginal maximum entropy partitioning yields asymptotically consistent probability density functions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 414–417, 2001.

[33] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical systems and turbulence, Warwick 1980*, pp. 366–381, Springer, 1981.

[34] M. Ragwitz and H. Kantz, “Markov models from data by simple nonlinear time series predictors in delay embedding spaces,” *Physical Review E*, vol. 65, no. 5, p. 056201, 2002.

[35] A. Serès, A. A. Cabaña, and A. A. Arratia Quesada, “Towards a sharp estimation of transfer entropy for identifying causality in financial time series,” in *Proceedings of the 1st Workshop on MIning DAta for financial applicationS (MIDAS 2016), Riva del Garda, Italy, September 19-23, 2016*, pp. 31–42, CEUR-WS. org, 2016.

[36] R. Hegger and H. Kantz, “Improved false nearest neighbor method to detect determinism in time series data,” *Physical Review E*, vol. 60, no. 4, p. 4970, 1999.

[37] J. Ameen and R. Basha, “Mining time series for identifying unusual sub-sequences with applications,” in *Innovative Computing, Information and Control, 2006. ICICIC’06. First International Conference on*, vol. 1, pp. 574–577, IEEE, 2006.

[38] R. Basha and J. Ameen, “Unusual sub-sequence identifications in time series with periodicity,” *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 2, pp. 471–480, 2007.

[39] J. Ameen and R. Basha, “Higherrarchical data mining for unusual sub-sequence identifications in time series processes,” in *Innovative Computing, Information and Control, 2007. ICICIC’07. Second International Conference on*, pp. 177–177, IEEE, 2007.

[40] G. E. Hinton, “A practical guide to training restricted boltzmann machines,” in *Neural networks: Tricks of the trade*, pp. 599–619, Springer, 2012.

[41] S. Sarkar, S. Sarkar, N. Virani, A. Ray, and M. Yasar, “Sensor fusion for fault detection and classification in distributed physical processes,” *Frontiers in Robotics and AI*, vol. 1, p. 16, 2014.

[42] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[43] R. Bischoff, U. Huggenberger, and E. Prassler, “Kuka youbot-a mobile manipulator for research and education,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011.

[44] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, “The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing,” in

Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK), pp. 1–8, VDE, 2010.

[45] E. Guizzo and E. Ackerman, “The rise of the robot worker,” *IEEE Spectrum*, vol. 49, no. 10, 2012.

[46] J. F. Engelberger, *Robotics in practice: management and applications of industrial robots*. Springer Science & Business Media, 2012.

[47] S. Morante, J. G. Victores, and C. Balaguer, “Cryptobotics: Why robots need cyber safety,” *Frontiers in Robotics and AI*, vol. 2, p. 23, 2015.

[48] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, “An experimental security analysis of an industrial robot controller,” in *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 268–286, IEEE, 2017.

[49] A. Munawar, P. Vinayavekhin, and G. De Magistris, “Spatio-temporal anomaly detection for industrial robots through prediction in unsupervised feature space,” in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pp. 1017–1025, IEEE, 2017.

[50] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, “Multimodal execution monitoring for anomaly detection during robot manipulation,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 407–414, IEEE, 2016.

[51] K. Häussermann, O. Zweigle, and P. Levi, “A novel framework for anomaly detection of robot behaviors,” *Journal of Intelligent & Robotic Systems*, vol. 77, no. 2, pp. 361–375, 2015.

[52] F. Baghernezhad and K. Khorasani, “Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots,” *Neurocomputing*, vol. 171, pp. 335–346, 2016.

[53] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," *arXiv preprint arXiv:1602.07109*, 2016.

[54] S. Windmann, S. Jiao, O. Niggemann, and H. Borcherding, "A stochastic method for the detection of anomalous energy consumption in hybrid industrial systems," in *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*, pp. 194–199, IEEE, 2013.

[55] S. Faltinski, H. Flatt, F. Pethig, B. Kroll, A. Vodenčarević, A. Maier, and O. Niggemann, "Detecting anomalous energy consumptions in distributed manufacturing systems," in *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, pp. 358–363, IEEE, 2012.

[56] A. Almalawi, X. Yu, Z. Tari, A. Fahad, and I. Khalil, "An unsupervised anomaly-based detection approach for integrity attacks on scada systems," *Computers & Security*, vol. 46, pp. 94–110, 2014.

[57] P. Guo, H. Kim, N. Virani, J. Xu, M. Zhu, and P. Liu, "Exploiting physical dynamics to detect actuator and sensor attacks in mobile robots," *arXiv preprint arXiv:1708.01834*, 2017.

[58] H. Wu, H. Lin, Y. Guan, K. Harada, and J. Rojas, "Robot introspection with bayesian nonparametric vector autoregressive hidden markov models," in *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pp. 882–888, IEEE, 2017.

[59] H. Wu, H. Lin, S. Luo, S. Duan, C. Xiang, B. Zhao, and J. Rojas, "Anytime, anywhere anomaly recovery through an online robot introspection framework," *arXiv preprint arXiv:1708.00200*, 2017.

[60] A. Brovelli, M. Ding, A. Ledberg, Y. Chen, R. Nakamura, and S. L. Bressler, “Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by granger causality,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 26, pp. 9849–9854, 2004.