Anomaly Detection on Time Series Data

Ipsit Pradhan

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Anomaly Detection on Time Series Data

Thesis submitted in partial fulfillment

of the requirements of the degree of

Master of Technology

Under the Dual Degree Programme

in

Computer Science and Engineering
(*Specialization*: *Information Security*)

by

Ipsit Pradhan

(Roll Number: 711CS2166)

based on research carried out

under the supervision of

Prof. Dr. Bidyut K. Patra

May, 2016

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Dr. Bidyut K. Patra
Professor

May 20, 2016

Supervisor's Certificate

This is to certify that the work presented in the dissertation entitled *Anomaly Detection on Time Series Data* submitted by *Ipsit Pradhan*, Roll Number 711CS2166, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Dr. Bidyut K. Patra

Dedication

This work is dedicated to my parents, my sister and my hometown of Rourkela.

Signature

Declaration of Originality

I, *Ipsit Pradhan*, Roll Number *711CS2166* hereby declare that this thesis entitled *Anomaly Detection on Time Series Data* presents my original work carried out as a postgraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" . I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 20, 2016
NIT Rourkela

Ipsit Pradhan

Acknowledgment

This research project would have been incomplete without the kind support of the individuals as well as this institute. I would like to thank all of them from deep of my heart.I would like to thank my supervisor **Dr. Bidyut K. Patra** Department of Computer Science and Engineering , NIT Rourkela for his great contribution and guidance throughout the project. He helped me with the topic by providing valuable ideas that helped me in completing the project. It is indeed a great privilege to be associated with Prof .**S K Rath** ,HOD ,Department of computer science and engineering . I am obliged to all the professors of the Department of Computer Science and Engineering, NIT Rourkela for helping me the basic knowledge about the field that greatly benefited me while carrying out the project and achieving the goal. I thank the staff of National Institute of Technology Rourkela for extending their support whenever needed. I would also like to thank Suratna Budalakoti, alumnus of National Institute of Technology Rourkela, for his guidance regarding the research area.

April 20, 2016
NIT Rourkela

Ipsit Pradhan
Roll Number: 711CS2166

Abstract

Anomaly detection is an important problem that has been researched within diverse application domains. Detection of anomalies in the time series domain finds extensive application in monitoring system status, mal-ware/spam detection, credit-card fraud etc. In this work we explore methods to detect anomalies in multivariate as well as uni variate time-series and proposed a novel method using Dictionary Learning, Sparse Representation, Singular Value Decomposition and Topological anomaly detection(TAD). We have tested the proposed method on real as well as synthetic data sets. Our novel method brings down the false positive rates as compared to the existing methods.

Keywords*: *Anomaly Detection*; *Time-series*; *Dictionary Learning*; *Electricity Theft Detection*; *Unsupervised Techniques.

Contents

Supervisor's Certificate ii

Dedication iii

Declaration of Originality iv

Acknowledgment v

Abstract vi

List of Figures ix

List of Tables x

1 Introduction 1

1.1 What are anomalies? . . . 2
1.2 Challenges . . . 2
1.3 Challenges in Discrete Sequence or Time Series . . . 3
1.4 Applications of Anomaly Detection on Discrete Sequences . . . 4
1.5 Motivation and Objective . . . 4
1.6 Problem Statement . . . 6
1.7 Thesis Organization . . . 6

2 Literature Review 7

2.1 Checking for anomalies as compared to an existing sequence database . 7
2.1.1 Kernel Based Techniques . . . 8
2.1.2 Window Based Techniques . . . 9
2.1.3 Using Sparse Coding and Latent Semantic Analysis . . . 10
2.1.4 Topological Anomaly Detection . . . 13
2.1.5 Markovian Techniques . . . 14
2.1.6 Hidden Markov Models(HMM) Based Techniques . . . 17
2.2 Detecting Anomalous subsequences Within a Long Sequence . . . 17

2.3 Determining If the Frequency of a Query Pattern In A Given Sequence Is Anomalous W.R.T Expectation . . . 18
2.3.1 Basic Approach to Solve the above problem formulation . . . 18
2.4 Conclusion . . . 18

3 A Method To Decrease False Positive Rate of Anomaly Detection 20
3.1 Introduction . . . 20
3.2 Part I: Sparse Representation of the Time Series with Latent Semantic Analysis . . . 20
3.3 Part II : Topological Anomaly Detection . . . 22
3.4 Combining Part I and II . . . 23
3.5 Experimental Results and Analysis . . . 24
3.5.1 Preparing the data . . . 24
3.5.2 Test Results . . . 25
3.6 Conclusion . . . 25

4 Conclusion and Future Work 30
4.1 Summary of Contributions of the Thesis . . . 30
4.2 Future Work . . . 30

References 32

List of Figures

1.1 A simple example of anomalies in 2-dimensional data 2
1.2 A time series showing the normal ECG0606 data set pattern of a person 4
1.3 Time Series with anomalies marked in blue 5

2.1 Analogy between a term-document matrix and a sparse feature matrix. 12
2.2 An Example of TAD with r = 0.1 and p = 0.1 15
2.3 An Example of TAD with r = 0.3 and p = 0.1 16

3.1 Schematics of our proposed work 23
3.2 Consumption pattern Time Series plot for House Number 1001 for 24 hours over 535 days 24
3.3 Pairs plot showing anomalies in House Number 1001 26
3.4 Visualization of Anomalies in house 1001 26
3.5 ROC for DL with LSA 28
3.6 ROC for TAD 29
3.7 ROC for our proposed method 29

List of Tables

3.1 Topological Anomaly Detection 27
3.2 Dictionary Learning with Latent Semantic Analysis 27
3.3 TAD + DL 28
3.4 Result Summary 29

Chapter 1

Introduction

The problem of anomaly or outlier or novelty Detection implies finding of patterns that do not adhere to an expected behaviour. These non-conforming patters are often referred to as discordant observations, outliers, anomalies, exceptions, aberrations, peculiarities and contaminants in different application domains. Of these, outliers and anomalies are two terms used most commonly. Anomaly detection finds immense use in a wide variety of applications including but not limited to credit card fraud detection, insurance, health care, cyber security and military surveillance.

Anomaly detection is critical because of the fact that it gives out actionable intelligence in a wide range of applications. A few examples of such application are anomalous traffic pattern monitoring in a computer network indicates a hacked system sending out sensitive information to unauthorized locales. Anomalous MRI images indicate presence of tumors. Anomalies in credit card usage pattern indicates identity theft or anomalous readings from space craft sensors could signify faults in components.

Detection of outliers has been a topic of great interest amongst the statistics community since the early 19^{th} century. Overtime many domain specific anomaly detection techniques have been developed by various researchers. Many are domain specific while others are more generic.

Sequences are ordered series of events. Sequences can be discrete, binary and continuous type, context specific to the type of events that form the sequence. Discrete and continuous are two of the most common sequences encountered in real life.[1]

Sequence data is found application domains such as bio-informatics, intrusion detection[2], healthcare, etc. Hence anomaly detection for sequence data becomes a topic of high interest among researchers. There is extensive work on techniques that differentiate novelty objects from other objects categorized as normal. [3][4].

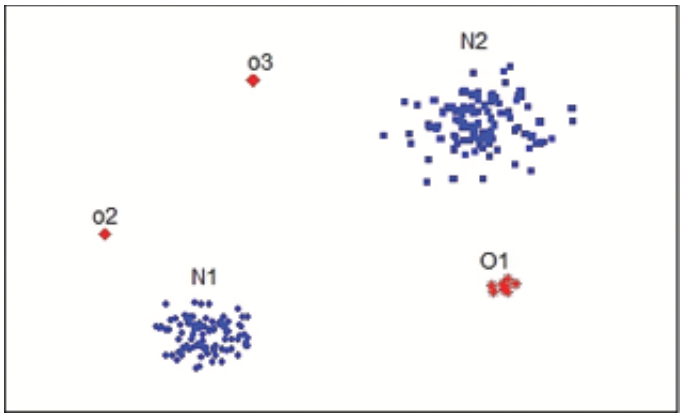


Figure 1.1: A simple example of anomalies in 2-dimensional data

1.1 What are anomalies?

1.1 shows simple two dimensional data. The data has two regions classified as normal, N_1 and N_2 as a major part of the observations lie in this region. Points that lie sufficiently further off these two regions are classified as outliers e.g. points O_2 and O_3 and the points in region O_1. Data might contain anomalies for various reason such as malicious activity, e.g. terrorist activity ,credit card fraud or system breakdown. The real life relevance of the anomalies is an important feature for analysts.

Anomaly detection is closely related to noise removal and novelty detection. Noise removal deals with removal of that data which is of no interest to the analyst. It is usually a part of pre-processing and cleaning the data before analysis. Novelty detection aims to find data that was previously missed.

1.2 Challenges

A simple approach to anomaly detection often involves defining a normal region and declaring any observation that does not conform to the normal behaviour as anomalous. But many factors make the approach challenging :

- Defining a region as normal, which exhaustively includes all possible normal behaviour is difficult. Along with that it is difficult to define boundaries between normal and anomalous behaviour. A data-point classified as anomalous can actually be a normal observation and or a data point classified as normal be anomalous.
- When malicious activities cause anomalies, the adversaries often adapt new methods to appear as normal
- In many domains normal behaviour keeps evolving and it becomes a difficult task to define normal.

- The exact notion of anomaly varies according the domains. For example in health care a small deviation can be termed as anomaly while large fluctuations in stock market can be considered as normal.
- Availability of labelled data for validation/training to be used by the technique
- The data often contains noise which make it difficult to separate it from the outliers.

Researchers have adopted methods from various fields such as machine learning, statistics, data mining, information theory, spectral theory etc. and have applied them to specific problem statement[3].

1.3 Challenges in Discrete Sequence or Time Series

Discrete and continuous are two of the most common sequences encountered in real life[1].

A time series is a sequence of data points made:

- over a continuous time interval.
- out of successive measurements across that interval.
- using equal spacing between every two consecutive measurements.
- with each time unit within the time interval having at most one data point

Sequence data is found application domains such as bio-informatics, intrusion detection[2], healthcare, etc. An example can be seen from 1.2. Hence anomaly detection for sequence data becomes a topic of high interest among researchers. There is extensive work on techniques that differentiate novelty objects from other objects categorized as normal.

Detection of anomalies in discrete sequences is a tough task since it involves exploiting the sequential nature of data. Below are some of the specific challenges :

- Anomalies within sequences have multiple definitions; an event or a sub sequence within a sequence might be anomalous. Each definition need to be handled carefully. A technique that can detect anomalies within a sequence might not be directly applicable to detect anomalies caused by a sub sequence of events occurring at once.
- Lengths of anomalies within sequences usually vary significantly across domains. Techniques highly rely upon the lengths defined by users which may or may not be optimal.

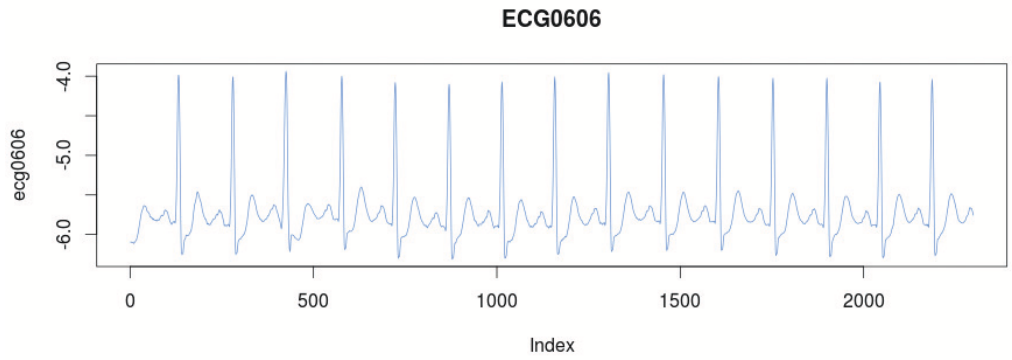


Figure 1.2: A time series showing the normal ECG0606 data set pattern of a person

- Since sequences can be long and alphabet sizes large computational complexity becomes a major issue.

1.4 Applications of Anomaly Detection on Discrete Sequences

A few applications of detection of anomalies on time series and discrete sequence are :

- *Operating System Calls/User Commands* Sequences are defined by an exhaustive list of all possible system calls or user commands. Deviations in such data usually correspond to "break-ins" in the computer system viruses or malicious users.
- *Biological Sequences such as DNA* Nucleic Acid or Protein bases correspond to symbols in the alphabet for such sequences. Detected anomalies for such sequences imply diseases or mutations
- *Sensor Data from Operational Systems* This is data collected through multiple discrete sensor system. These data sets typically have a large alphabet size. Fault scenarios or accidents are implied when anomalies are detected in such sequences.
- *Navigational Click Sequences From Websites* Anomalies in such data indicates unauthorized access or malicious behaviour.

1.5 Motivation and Objective

Anomaly Detection on time series finds real world application in a diverse range of fields as mentioned in the previous section. As pointed out in challenges in the previous section we can see that anomaly detection on time series is not only computationally

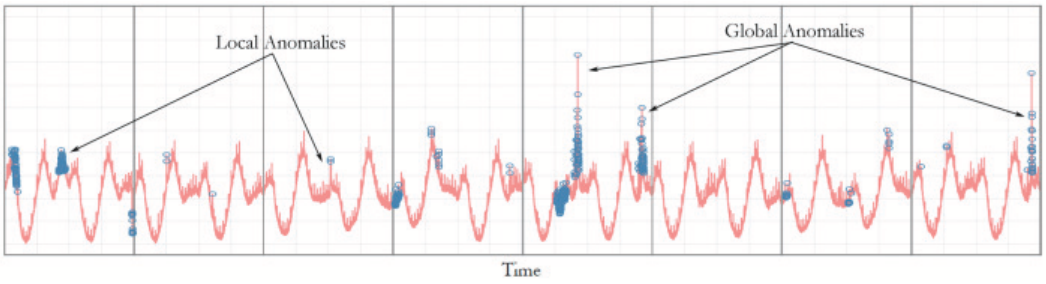


Figure 1.3: Time Series with anomalies marked in blue

intensive but also difficult because of the variable alphabet size and unavailability of marked data sets for training purposes.

Detection of anomalies is followed by investigation and mitigation of the cause of anomalies. Every time an anomaly is detected in sensor data from a factory someone has to manually inspect the fault location. Often false alarms are raised by such anomalies which lead to unnecessary expenditure of man power and time. While it is relatively easy to just decrease the false positive rate , it is important to keep the recall rate as high as possible.

Dependence upon an accurately marked data set can be eliminated using unsupervised and semi-supervised anomaly detection techniques. Researchers adopt multiple approaches to solve the same problem creating filters which bring down the false positive rate while maintaining or minimally decreasing the recall rate/accuracy. Our main objectives are:

1. To investigate techniques for detection of anomalies in discrete sequences, especially time-series.

2. To device a novel method to detect anomalies from unmarked data sets.

3. To decrease the false positive rate with respect to existing techniques while maintaining or improving the recall rate.

4. Implementation and testing of our method on real-world data.

1.6 Problem Statement

The main goal of this work is to detect anomalies in discrete sequences and time series data using techniques that do not require a labelled data set to train a model. We obtain the most apt data sets for the study of the same. We use supervised and unsupervised techniques to obtain anomalous data points/sub sequences/sequences from the time series. Optimize existing techniques which decrease the computational load and improve upon the existing methods. Do a comparative analysis of the novel technique with the existing methods and check the improvement.

1.7 Thesis Organization

The present thesis is organized into **six** chapters. ***Chapter 1*** presents introduction to anomaly detection and its challenges. ***Chapter 2*** presents a literature review on Anomaly detection techniques for time series data and problems related to various approaches. ***Chapter 3*** discusses in detail, Dictionary learning and sparse representation, presenting a few proposed changes to the existing method, the unsupervised technique: Topological Anomaly Detection (TAD) and its modification relevant to out data set. ***Chapter 4*** concludes the work done, highlighting the contributions and suggests possible future work.

Chapter 2

Literature Review

Anomaly detection techniques for discrete sequences can be denoted under 3 broad problem statements. The problem statements can be generalized with the following three scenarios:

- *Scenario 1:* To check if a given sequence is anomalous with respect to a database of sequences.
- *Scenario 2:* To detect subsequences with are anomalous within a lengthy sequence.
- *Scenario 3:* To detect whether the frequency of a occurrence of a particular sequence is very much different from what is expected.

In the following sections we discuss the existing work regarding each scenario.

2.1 Checking for anomalies as compared to an existing sequence database

This is the most available problem statement found as compared to the other two scenarios. One application of this scenario is when a security analyst want to check whether there has been any access of the system by an unauthorised user, he refers to the past normal sessions to check for the deviation. Most existing work on this problem assign a score of abnormality which ranks the sequences and determines the most anomalous ones.

This problem statement has two types of variations. The first variant is assumed to contain a database consisting only of normal sequences. The second variant uses unsupervised techniques to find anomalies from a database with no labels. For the latter it's assumed that only a minority of points are anomalies.

The first problem formulation variant uses semi-supervised techniques and can be stated as follows:

Definition 1: Given n sequences, $\mathbf{S} = s_1,s_2,...,s_n$, and s_{test} is an element from the test set,S_q. Compute the degree of anomaly of s_{test} w.r.t $\mathbf{S}$.

Lengths of sequences in $\mathbf{S}$ and S_q may vary. Additional tests are required post assignment of anomaly scores, to determine whether the score is significant enough to term the observation as an anomaly or not.

The semi-supervised problem can be formulated as :
Definition 1a: : Given a set of sequences $\mathbf{S} = s_1,s_2,...,s_n$, find all the sequences in S which when compared to rest of S , are anomalous. Methods to solve the formulation in *Definition 1* usually take two steps to operate. The first steps involve learning a model that represents normal behaviour. The second step calculate the likelihood of the test pattern being generated using the learned model In the following subsections we discuss the anomaly detection techniques based on the way unit test sequences are analyzed.

2.1.1 Kernel Based Techniques

Such techniques calculate similarity, pairwise, among sequences using similarity measures and then point based algorithms to detect anomalies. In basic kernel techniques first a pairwise similarity matrix is calculated for all the sequences in training set in S. Then S_q is matched against the matrix to get an anomaly score.

Using Different Point Based Algorithms To Detect Anomalies

k-Nearest Neighbour [5] and and clustering based [6] are two point based algorithms to detect anomalies. In [7] proposed by Budalakoti et al, is based on a clustering techniques where the training sequences are categorized into a fixed set of clusters with k-medoid. Then the anomaly score of the test sequence is calculated as the inverse of its closeness to its nearest medoid. Stochastic clustering techniques that do not explicitly require a similarity matrix for finding clusters have also found use in anomaly detection. An example is the representation of probabilistic suffix trees as clusters by Yang et al [8]. Mixtures of HMMs and Maximum Entropy Models are among other stochastic techniques.

Using Different Similarity Measures

Simple Matching Coefficient (SMC) is the most basic similarity measure that is used to compare pairs of discrete sequences. SMC uses the count of the number of positions in which two are exactly the same. The major disadvantage of this method is that it requires the two sequences to be of the exact same length.

Many techniques use the common subsequence of the longest size as its similarity measure as can calculate similarity for two sequences even when the lengths do not

match. A setback for using LCS is its large computational complexity involved who order of magnitude much higher than that of SMC.

Advantages and Disadvantages of Kernel Based Techniques. The major advantage once a similarity kernel is obtained, there can be the application of any similarity based technique. Techniques involving calculation of similarity can take advantage of existing work on sequence similarity and apply Clustering or Nearest neighbour algorithm.

High dependence on the similarity measure is a major disadvantage for the kernel based techniques. Another major disadvantage is that they have high computational complexity.

2.1.2 Window Based Techniques

Overlapping sequences of fixed length are extracted for the test set in these techniques. Exact extracted window is given an anomaly score. Then anomaly score of all the windows are taken into consideration and aggregated to get the anomaly score of the entire sequence.

The utility of window based techniques can be understood by examining the shortcomings of Kernel Based Techniques. The latter estimates $\mathrm{P}(S_q|M)$, which is the conditional probability of existence of the entire sequence S_q given a learned model M. In the research community it has often been argued that the cause of an anomaly can be pin pointed to shorter sequences within a large sequence [9]. Analysing the entire sequence as a whole may lead to skipping of the anomalies which may not be easily distinguished from the existing variation.

A conventional technique to extract windows is by sliding a window of fixed length along the sequence. The extracted windows are denotes as w_1,w_2,...,w_t and each element of a particular window can be referred to as $w_t[i]$.

Assuming that a sub sequence a_q is contained in S_q, is an actual cause of anomaly. If k, is the length of the window, the cause of the anomaly will occur partly or wholly in $|a_q| + k - 1$ windows. Hence we can detect anomalies by detecting at least once window like this.

A very crude window based technique works in the following manner. During the training phase, sliding windows of length k are extracted from all sequences in the training set and their frequency if maintained in a "normal repository". In the test phase we extract windows using the same method as in the training phase.Each window W_i is assigned a likelihood which is proportional to the frequency of the sequence that has been saved in the repository. A threshold value λ is set to determine whether the extracted window is anomalous or not. Let $L(W_i)$ be the likelihood of the window, if $L(W_i) \geq \lambda$ it is categorized as an anomaly or vice-versa.

Advantages and Disadvantages of Window Based Techniques A major advantage of using window based techniques as compared to kernel based techniques is that the former can detect anomalies confined to a smaller region within the longer sequence. It is fairly simple to construct dictionaries consisting of normal values which can be further optimized with the use of the right data structures.

Window based technique depend heavily upon the value of k, the size of the extracted window, this is a major disadvantage of using window based techniques. If the value k is very small, majority of the k-length windows would have a high chance of occurring in training data on the other hand if the value of k is very large the chance of occurring in training data will be fairly low. Thus in both the cases unless the value of k is tuned to optimality the ability to differentiate between anomalous and non-anomalous sequences would be highly limited. Another disadvantage is that storing all the unique windows and their frequencies would consume a large amount of memory space.

2.1.3 Using Sparse Coding and Latent Semantic Analysis

There have been numerous proposals for anomaly detection on uni-variate as well as multivariate time-series. On of them deals with discrete sequences as a set of independent observations that are in a high-dimensional space. Since the data can be converted to a a lower dimensional subspace we can find anomalous points by observing the deviation from this subspace. This method can capture interdependency among multiple variables but does not consider time-domain correlations. Principal Component Analysis (PCA), which is a dimension reduction technique, is the most elementary technique to identify the subspace. Another approach to the same problem is to estimate the models that generate the series such as Vector Autoregressive models (VAR) and state space models(SSMs).

The first part of this semi-supervised method which requires the labels of normal classes only during training. This method consists of two stages; first is feature extraction using sparse representation and second is learning relationship with reduction of dimensionality.

Dictionary Learning

[10] is an example of classical dictionary technique which consists of a training set $\mathbf{X} = [x_1,....,x_n]$ in $\mathbb{R}^{m\times n}$ for the cost function :

$$f_n(\mathbf{D}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(x_i, \mathbf{D}) \tag{2.1}$$

where $\mathbf{D}$ in $\mathbb{R}^{m\times k}$ is a dictionary, every column represents a particular basis vector, l is

a loss function so that l(x,**D**) is minimized if **D** is efficient at representing the signal. This loss function can be defined as the optimal value of a l_1-*sparse-coding* problem as in [11] :

$$l(\mathbf{x}, \mathbf{D}) \triangleq \underset{\alpha \in \mathbb{R}^k}{min} \frac{1}{2}\|x - D\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{2.2}$$

where λ is a parameter to regualarize the equation. Such a problem is also known as *Lasso* or *Basis Pursuit* [12]. As we know that l_1 penalty gives a sparse solution for α, there is no analytic link between the value of λ and the corresponding effective sparsity and in order to prevent **D** from being arbitrarily large (which would lead to arbitrarily small values of α) it is common to constrain its columns $(\mathbf{d}_j)_{j=1}^k$ to have l_2 norm less than or equal to one [11]. C is a convex set of matrices verifying this constraint :

$$C \triangleq \{\mathbf{D} \in \mathbb{R}^{m \times k} s.t. \forall j = 1, ..., k, \mathbf{d}_j^T \mathbf{d}_j \leq 1\} \tag{2.3}$$

Though the problem to minimizing the empirical cost of f_n(D) isn't convex w.r.t D. It can be re-phrased as a con-joined optimization problem w.r.t **D** and coefficients α=[α_1,...,α_n] of the sparse decomposition, which is convex w.r.t two variables **D** and α when either one is fixed :

$$\underset{D \in C, \alpha \in \mathbb{R}^{k \times n}}{min} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2}\|x_i - D\alpha_i\|_2^2 + \|\alpha\|_i \right) \tag{2.4}$$

An intuitive way to solve this problem is to alternately keep one variable fixed and solve for the other one, minimizing over multiple iterations. As used by [13] dictionary learning consists of sequences of updates:

$$D_t = \prod_C \left[D_{t-1} - \frac{\rho}{t} \nabla_D l(x_t, D_{t-1}) \right] \tag{2.5}$$

Sparse Representation Using Learned Dictionary

Let $\mathcal{D}$ = (d_1,...,d_n) be a basis dictionary learned from the last section. It consists of a set of bases b_j (j=1,...,m) of the signal. We obtain sparse representation using the following optimization :

$$\underset{X}{minimize} \|\mathcal{Y} - \mathcal{D}\mathcal{X}\|_2^2 + \lambda \sum_{i=1}^{n} \|x_i\|_1 \tag{2.6}$$

where $\mathcal{Y} \in \mathcal{R}^{m \times n}$ is the signal matrix and $\mathcal{X} \in \mathcal{R}^{m \times n}$ is the matrix of sparse representations. We run a sliding window over the time series and obtain sets of non-overlapping sub-sequences. $\mathcal{S}^{(k)}$=($s_1^{(k)}$,...,$s_n^{(k)}$), where $s_t^{(k)}$ is a subsequence of time-series $\mathcal{T}^{(k)}$ that begins at ⊔=1.

In the training phase we run (3.4),(3.5) and (3.6) iteratively over $\mathcal{Y}_{ref}$ = $\mathcal{S}_{ref}^{(1)}$,...,$\mathcal{S}_{ref}^{(n)}$ to obtain sparse representation $\mathcal{X}_{ref}^{(k)}$ and optimized dictionary $\mathcal{D}_{ref}^{(k)}$. Then

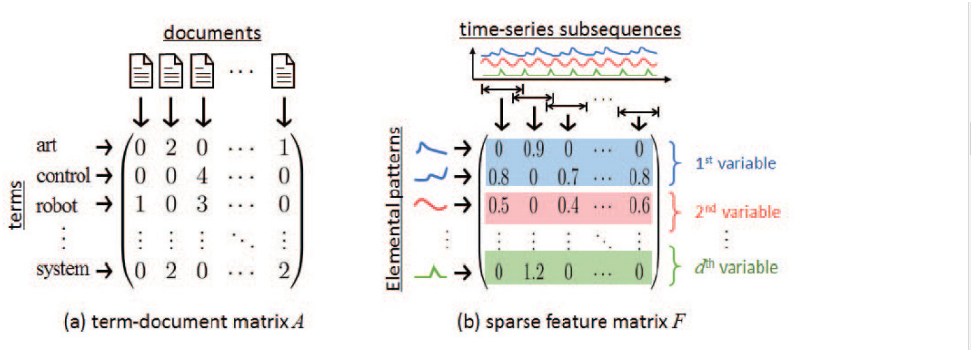


Figure 2.1: Analogy between a term-document matrix and a sparse feature matrix.

in test phase $\mathcal{X}_{test}^{(k)}$ is found using the fixed dictionary $\mathcal{D}_{ref}^{(k)}$. In order to treat the d-variable time-series we stack all the extracted sparse features for d variables:

$$F = \begin{Bmatrix} \mathcal{X}^{(1)} \\ \mathcal{X}^{(2)} \\ \cdots \\ \mathcal{X}^{(k)} \end{Bmatrix}$$

Relationship Learning with Latent Semantic Analysis

After feature extraction of $\mathcal{T}_{ref}$, the co-occurence relations of the local patterns are known. Our idea is based upon Figure 2.1. In a term document matrix element (i,j) denotes the frequency of the i^{th} term in the j^{th} document. In Natural Language Processing(NLP), co-occurence analysis from term document matrices is done with a dimensionality reduction technique LSA or Latent Semantic Analysis [14]. LSA begins with Singular Value Decomposition (SVD):

$$F = U \sum V^T \tag{2.7}$$

$$\hat{F} = \sum^{-1} U^T a \tag{2.8}$$

$$\tilde{F} = U \sum a \tag{2.9}$$

Using the analogy from figure 2.1 we utilize the same technique to extract pattern relations of the local time-series. In training as well as test phases, the time-series $\mathcal{T}_{ref}$ and $\mathcal{T}_{test}$ are transformed to sparse feature matrices $\mathcal{F}_{ref}$ and $\mathcal{F}_{test}$ respectively. We apply (3.7) to $\mathcal{F}_{ref}$ in training phase to get U_{ref} and $\sum_{ref}$. In the test phase the the feature matrix $\mathcal{F}_{test}$ is transformed into semantic space and then reconstructed into original space by using (3.8) and (3.9) respectively.

Because the rank-reduced matrices $U_{ref}^{(k)}$ and $\sigma_{ref}^{(k)}$ preserve only the essential latent semantics, (3.9) cannot reconstruct the original feature perfectly and produces

reconstruction errors, If the latent semantics (co-occurrence relations) in the test time-series data are not different from those in the reference data, the reconstruction errors will be small [15]. If the data has anomalies the errors during reconstruction will remain large. Hence to calculate the anomaly score we used the square of the reconstruction error.

$$(AnomalyScore) = (F - \tilde{F}) \circ (F - \tilde{F}) \tag{2.10}$$

where $\circ$ is entrywise product.

2.1.4 Topological Anomaly Detection

Topological Anomaly Detection is a relatively recent approach which improves upon the performance of existing algorithms, like RX on hyper-spectral datasets. TAD is used in [16] to find anomalies in polarimetric images. We used the same heuristics in case of time-series datasets.

Let X be a finite number of points in vector space $\mathbb{R}^k$. Typically we assume X to be around 1 million points and k, 200. r is a positive real number. G_r is a graph with vertex set as X s.t. there exists and edge between x and y such that the distance between x and y is smaller than r. r is denoted as the resolutions of graph G_r. We consider two points with distance less than r to be indistinguishable hence we name r as the resolution of the graph. G_r can be thought of as a graph obtained by placing an edge from x is y is inside a ball of radius , with centre at x.

In case of time-series, when large number points are considered indistinguishable we assume that the point amounts to background points and all non background points as anomalies. To be specific let p be defined as a percentage $\in (0, 100)$, which will be called background points. A component H of G_r is defined to be a background component when H consists of more than p percent of points of X. A point in the background is denoted as background point and points which are not in the background are denoted as anomalies. In practice, p is expected to be 1 percent approximately and roughly 95 percent of points in X are background points, however these values may vary from a case to case basis. The degree of an anomaly is defined by the values of distance between x and y, $d(x, y)$, the larger the distance larger is the degree of anomaly.

This technique differs from other kNN anomaly detection algorithms in the way that we set a maximum inter-observational distance instead of setting the parameter k. Here instead of using k as a tuning parameter we use p to tune. The number of points classified as anomalies depend directly upon the value of p. This is shown in figure 2.2 and figure 2.3. The datasets used in Figure 2.2 and Figure 2.3 are taken

from Google Finance Domestic Trends[1].

2.1.5 Markovian Techniques

These category of techniques learn a model using the training set sequences. The model is used as an emulation of the actual distribution which generates observations classified as normal. Usually the probability of a sequence is factorized using:

$$P(S) = \prod_{t=1}^{l} P(s_t|s_1, s_2, \ldots, s_{t-1}) \tag{2.11}$$

Where l is the sequence length and s_i is the symbol at position I in S.

The short term memory property of sequences is utilized by Markovian techniques. This property is observed across various domains. This property is essentially a higher-order Markov condition where it is stated that the conditional probability of occurrence of a symbol , given the sequence observed so far can be approximated as:

$$P(s_t|s_1, s_2, \ldots, s_{t-1}) = (s_t|s_{t-k}, s_{t-k-1}, \ldots, s_{t-1}) \tag{2.12}$$

There are two phases to Markovian techniques, training and testing. In the training phase a probabilistic model is learned using **S**. In the testing phase the conditional probability of each sequence is calculated using (2.2). The assigned final anomaly score is the inverse of the probability. There are three kinds of Markovian techniques:

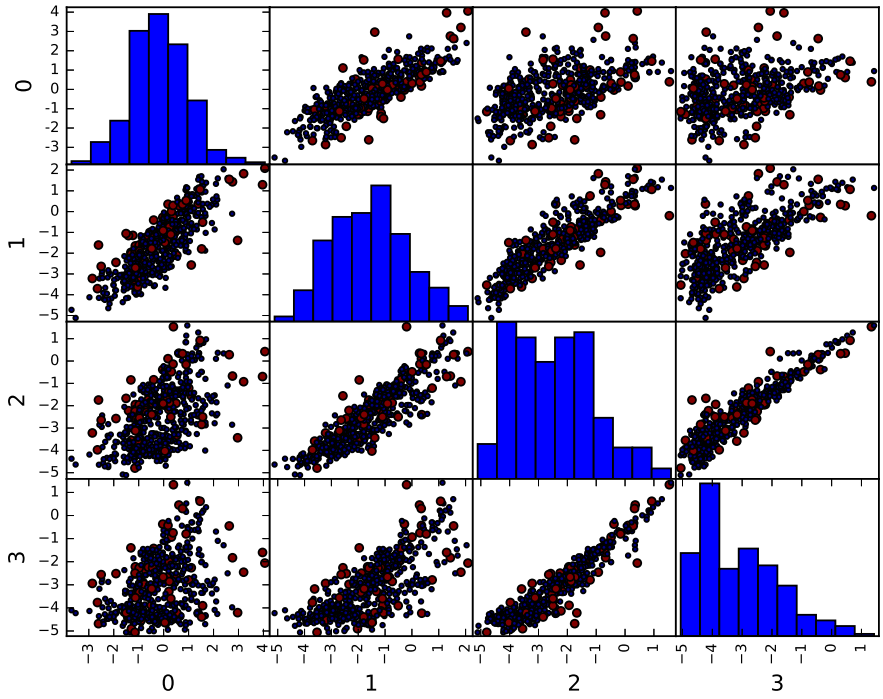
i. *Fixed Markovian Techniques :* In these techniques the length of history is fixed to k. This history is used to calculate the computational probability of a symbol. Different variants of this technique has been proposed.

ii. *Variable Markovian Techniques :* To overcome the shortcomings of fixing the value of k in Variable Markovian Techniques this technique is used. Variable Markovian Techniques solve this problem by allowing symbols to be conditioned according to a variable length of k.

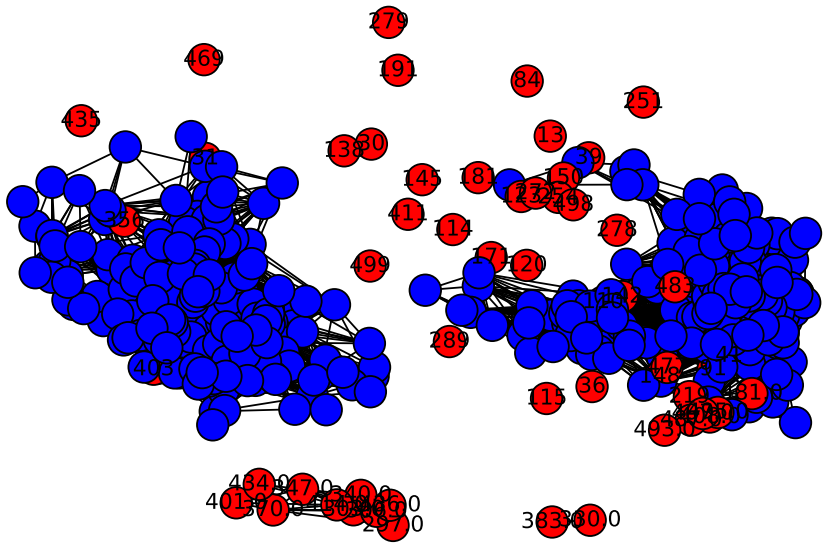
iii. *Sparse Markovian Techniques :* Sparse Markovian techniques are more flexible than the previous two techniques. The estimate the conditional probabilities based on the preceding k symbols are aren't necessarily continuous.

Advantages and Disadvantages of Markovian Techniques The major advantage of using Markovian technique is that each event is analyzed w.r.t its immediate context. Hence such techniques can detect anomalies even if they are localized. Sparse and variable markovian technique provide flexibility regarding the size of context history that is observed for every symbol. Hence if in a normal symbol

[1]https://www.google.com/finance/domestic_trends?ei=mOBCV4DBFIOe0ATjroWQBw

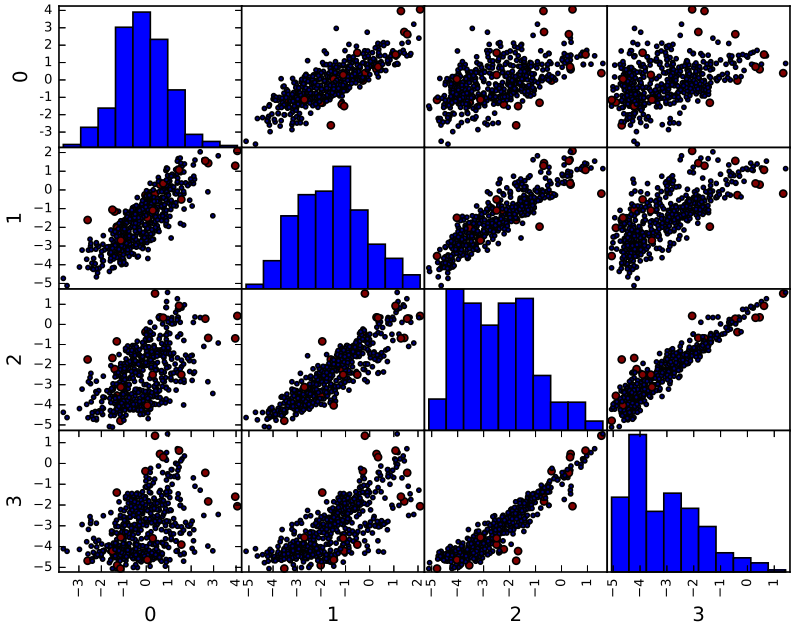


(a) Pairs plot with 4 features

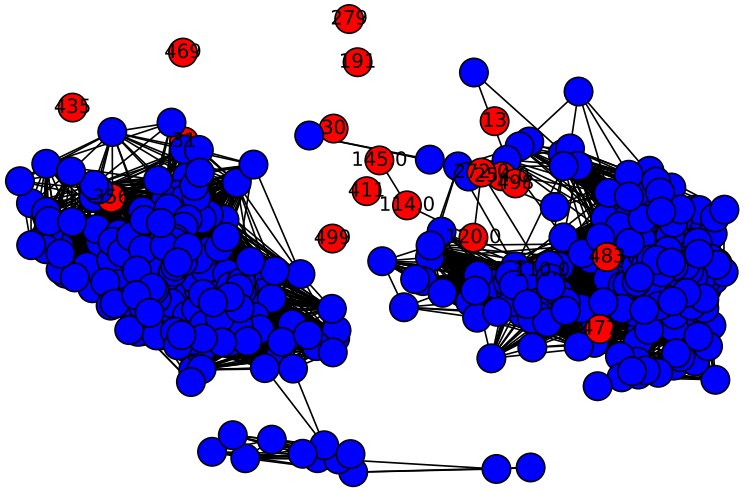


(b) Graphical Visualization

Figure 2.2: An Example of TAD with r = 0.1 and p = 0.1



(a) Pairs plot with 4 features



(b) Graphical Visualization

Figure 2.3: An Example of TAD with r = 0.3 and p = 0.1

sequence, the chances of observing a symbol w.r.t. its k-length history, using sparse and variable markovian technique we can approximate using a shorter history where the symbol would have a higher probability of occurrence. Such techniques help bring down the false positive rate.

Notwithstanding the advantages there are certain disadvantages to markovian techniques. Probability of *truly* anomalous symbols will be magnified since it will be tuned with a shorter context history, on the other hand, in fixed Markovian technique will give a low probability. The other two Markovian techniques have high false negatives.

2.1.6 Hidden Markov Models(HMM) Based Techniques

HMMs are strong finite state machines which are widely used to model sequences [17] and well as detect anomalies in sequences. These techniques transform input sequences to a state space that is hidden. The intuition behind using these techniques is that the basis of sequences is captured by the hidden space.

Advantages and Disadvantages of HMM Based Techniques If the assumption behind the hidden state are accurate, the transformed data will detect anomalies with better accuracy.

Initializing the HMM is not always intuitive, and bad choice for these initialization amounts to sub optimal performance.

2.2 Detecting Anomalous subsequences Within a Long Sequence

Techniques that come under the solution to this category of problem formulations:

Definition 2: Detect short sequences which are anomalous with respect to a long sequence T.

This definition is very generic to several domains where activities are observed over a long time. An example is fraud detection in credit card where, where electronic transactions of individuals are tracked and an anomalous discord may indicate misuse/theft.

A very basic technique to solve this problem is as follows : To begin with, all windows of length k are extracted from the sequence under consideration T and stored in form of a database of windows of fixed length, denoted as T_k. Each window is compared to rest of the database and assigned an anomaly score. Windows with anomaly score above the user defined threshold are termed anomalous. This technique is the core of a numbers of works by Keogh et al. They were originally presented with

regards to time series data only but this can be easily extended to discrete sequences as well.

A major hindrance with this basic technique is that when a window is compared to other overlapping extracted sequences they will be highly similar. Hence if two anomalies overlap they may be similar and the anomaly score may not be high.

2.3 Determining If the Frequency of a Query Pattern In A Given Sequence Is Anomalous W.R.T Expectation

Methods solving this problem statement involve:

Given a small query sequence s, a long test pattern S and a training set $\mathbf{S}$, determine if the frequency of occurrence of$s$ in S is anomalous w.r.t to occurrence of s in $\mathbf{S}$.

Alternately this problem is also referred to as *surprise detection*.

2.3.1 Basic Approach to Solve the above problem formulation

An elementary technique to solve the problem in the section involves assigning an anomaly score for the given query test pattern s as follows: Find the number of time the query pattern has occurred in S and $\mathbf{S}$. The anomaly score for s is calculated as the difference between frequency of s in S and the expected frequency of s in any sequence in $\mathbf{S}$.

$\hat{f}_S(s)$ is frequency of occurrence of the query pattern in the long test sequence S

$$\hat{f}_S(s) = \frac{f_S(s)}{|S|} \tag{2.13}$$

$\tilde{f}_{\mathbf{S}}(s)$ is frequency of occurrence of the query pattern in the long test sequence S

$$\tilde{f}_{\mathbf{S}}(s) = \frac{1}{|\mathbf{S}|} \sum_{\forall S_i \in \mathbf{B}} \frac{f_S(s)}{|S|} \tag{2.14}$$

The final anomaly score is computed as below:

$$A(s) = |\hat{f}_S(s) - \tilde{f}_{\mathbf{S}}(s)| \tag{2.15}$$

2.4 Conclusion

From this chapter we conclude that there are 3 basic problems related to detection of anomalies on discrete sequences. There has been extensive work on this field, however setting the value of the window size k and storing unique word sequences are major

roadblocks to efficient functioning of algorithms. Many existing algorithms have a high false positive rate which must be decreased.

The next chapter provides a detailed description of our proposed method to decrease the False Positive Rate during anomaly detection in time series data and apply our algorithm to find tampering with smart electricity meter.

Chapter 3

A Method To Decrease False Positive Rate of Anomaly Detection

Two of the major parameters that we take into consideration while evaluating anomaly detection algorithms. When anomalies are detected each anomaly is usually investigated on a case by case basis which consumes time and effort. False positives lead to wastage of such effort and time. In this chapter we discuss a method to reduce the false positive rate.

3.1 Introduction

Our solution to the problem of decreasing the false positive rate of anomaly detection techniques involves a two part process. To give an overview, we intuitively use a semi-supervised technique and an unsupervised technique to detect anomalies individually and then use them both as filters to reduce the rate of false positives.

3.2 Part I: Sparse Representation of the Time Series with Latent Semantic Analysis

This is the first part of our solution which is again a three part process. The first part is learning a basis dictionary from the part of the dataset that has be taken as the normal data. We do this iteratively by optimizing :

$$\underset{X}{minimize} \|\mathcal{Y} - \mathcal{D}\mathcal{X}\|_2^2 + \lambda \sum_{i=1}^{n} \|x_i\|_1 \tag{3.1}$$

Algorithm 1 and Algorithm 2 show the process to create and optimize the basis dictionary. After we get an optimized dictionary and a sparse representation of the training data we begin Latent Semantic Analysis.

We use SVD to decompose the sparse representation into three components. The details of how anomalies are detected is discussed in Chapter 1. In our work instead of using all three components to reconstruct the test window we use only one parameter. This gives a satisfactory recall rate but a high false positive rate. Eliminating one parameter eliminates and entire matrix and hence reduces the number of calculations. This in order reduces the processing time of the algorithm.

Input: $x \in R^m \sim p(x)$(random variable and and algorithm to draw i.i.d samples of p), $\lambda \in R$(regularization parameter), $\mathbf{D}_0 \in \mathbb{R}^{m \times k}$(initial dictionary), T (number of iterations)

$A_0 \leftarrow 0$, $B_0 \leftarrow 0$(reset the "past" information);

for $t = 1$ *to* T **do**

Draw x_t from p(x);

Sparse coding : compute using Lasso Regression :

$$\alpha_t \triangleq \underset{\alpha \leftarrow R^k}{arg\ min} \frac{1}{2}\|x_t - \mathbf{D}_{t-1}\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{3.2}$$

$\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \alpha_t \alpha_t^T$;

$\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + x_t \alpha_t^T$;

Compute $\mathbf{D}_t$ using Algorithm 2, with $\mathbf{D}_{t-1}$ as warm restart, so that

$$\mathbf{D}_t \triangleq \underset{D \in \zeta}{arg\ min} \frac{1}{t} \sum_{i=1}^{t} \|x_t - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha\|_1 \tag{3.3}$$

end

return $\boldsymbol{D}_T$

Algorithm 1: Dictionary Learning

Input: D=$[d_1,...,d_k] \in \mathbb{R}^{m \times k}$, A=$[a_1,...,a_k] \in \mathbb{R}^{k \times k} = \sum_{i=1}^{t} \alpha_i \alpha_i^T$,
B=$[b_1,...,b_k] \in \mathbb{R}^{m \times k} = \sum_{i=1}^{t} \mathbf{x}_i \alpha_i^T$

Result: Updated Dictionary

while *There is no convergence* **do**

for *j = 1 to k* **do**

Update the j-th column to optimize (3.2)

$$\mathrm{u}_j \leftarrow \frac{1}{A_{jj}}(b_j - Da_j) + d_j \tag{3.4}$$

$$\mathrm{d}_j \leftarrow \frac{1}{max(\|u_j\|_2, 1)} u_j \tag{3.5}$$

end

end

Algorithm 2: Dictionary Update

3.3 Part II : Topological Anomaly Detection

The original algorithm for TAD is discussed in Chapter 1. In the original algorithm a user has to explicitly enter the value of graph resolution *i.e.* the inter-observational distance above which two points are considered not connected. In our work instead of explicitly using the distance the user has to set the percentile of distances as the graph resolution. This works very well when the exact threshold distance is unknown. When the distance between two points is more than rq percentile we do not connect the points using an edge. The parameter pct specifies the percentage of points in the dataset a point must be connected to, so that the point is classified as a background point.

Input: P=[p_1,p_2,...,p_k],r=Percentile Resolution Of The Graph,pct=Percentage for Background Point
Result: Anomalies
counter=0, Anomalies=[];
for *i = 1 to k* **do**
  for *j = 1 to k* **do**
    D_{ij} = Dist(p_i, p_j);
    if $D_{ij} \geq r$ **then**
      Do not setup edge between p_i and p_j;
    else
      Setup edge between p_i and p_j;
    end
  end
end
for *i = 1 to k* **do**
  for *j = 1 to k* **do**
    if p_i *is connected to* p_j **then**
      counter++;
    end
  end
  if *counter* $\geq$ *pct* **then**
    Anomalies.append(p_i)
  end
end

Algorithm 3: Topological Anomaly Detection

3.4 Combining Part I and II

Let $\mathcal{R}^{\mathcal{I}}$ be the set of anomalies detected from Part I and $\mathcal{R}^{\mathcal{II}}$ be the set of anomalies detected from Part II. The two parts use two completely different approaches to the same problem hence we intuitively combine the results of Part I and Part II of our method.

$$FinalResults = \mathcal{R}^{\mathcal{I}} \cap \mathcal{R}^{\mathcal{II}} \tag{3.6}$$

Hence we built a method that uses two filters that have different approaches to the problem. The next section provides details of our experiment using the methods and gives comparative statistics with respect to using each method individually. Figure 3.1 shows a flow chart for the processes to find anomalies using our proposal.

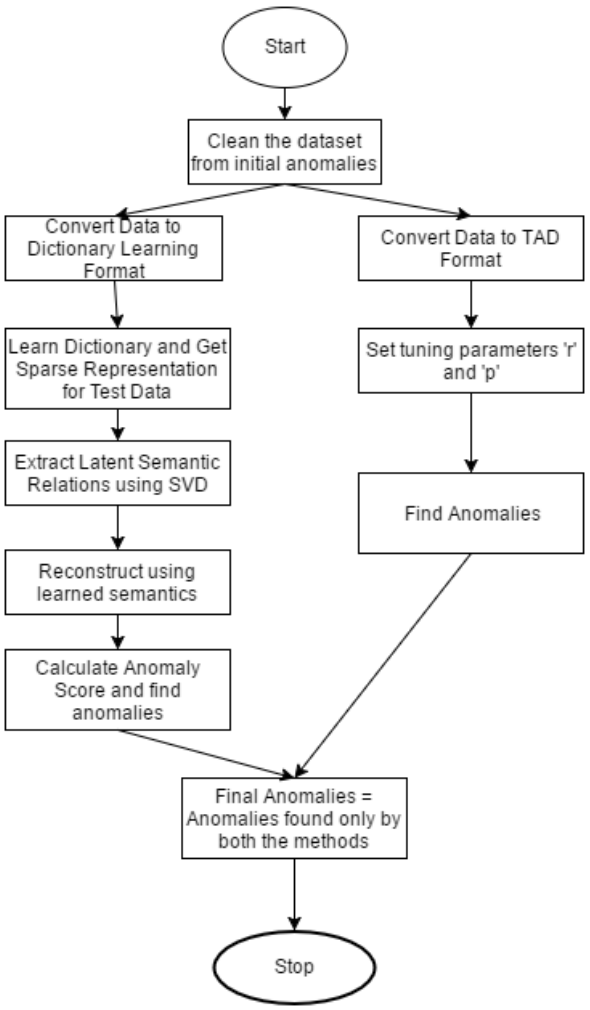


Figure 3.1: Schematics of our proposed work

3.5 Experimental Results and Analysis

The data set we used to test our heuristics is the electricity consumptions data released by ISSDA, Ireland on Electricity Consumption Survey using Smart Meters [1]. The data set contains the electricity consumption pattern of 5000 households throughout the day for 535 days. Each household's consumption is sampled every 30 mins which means there are 48 readings per day. We treat each half an hour reading as a feature , a visualization of house number 1001 is given in Figure 3.2. We implemented our algorithm in this data set to find tampering with the smart meter.

As mentioned in [18] there can be three types of attack on the Smart Meter :

- *Physical Tampering* : Users tamper with the internal mechanism of the the Smart Meter to report a lower than actual usage of the power supply. This leads to lower bills and is theft of electricity.
- *Cyber Attacks* : Malicious adversaries can tap into the communication link to affect the working of the smart meters.
- *Data Attacks* : This can be done through physical Tampering or Cyber Attack to report reading other than the actual ones.

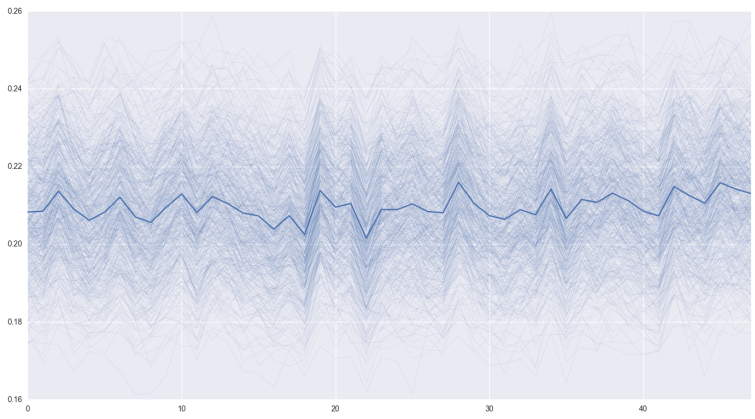


Figure 3.2: Consumption pattern Time Series plot for House Number 1001 for 24 hours over 535 days

3.5.1 Preparing the data

We ignore houses for which we do not have the readings for all 535 days or any missing values. In this work we assume that all the users who have volunteered have

[1] http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

not tampered with their AMI(Advanced Metering Infrastructure). We injected 10 artificial anomalous days' reading for each house. In the anomalies we injected we aimed to find those reading that were too high or too low as compared to the usual consumption pattern of the user. Though both the values indicate tampering with the smart meters, very high values indicate Data Attacks and very vow values indicate theft of electricity. We use the following general formulae to generate the anomalies:

i. Very low values = (median feature values) × random(0.001,0.008)

ii. Very High Values = (median feature values) × random(2.5,6) + random(1,3)

As TAD is an unsupervised method we do not need to separate the data into training and test sets but we need to separate the data for dictionary learning and latent semantics. Instead of setting the value of resolution of the graph explicitly for each house we use the percentile of distances(rq) and percentage of points to be connected with, to classify as back ground points as the tuning parameters.

For dictionary learning we randomly sample 100 days that we take as a reference or training set. Though there are 1000 houses in the data set we consider data only from 20 houses for our testing purposes. We set rq = 0.75 and p = 0.3.

3.5.2 Test Results

Table 3.1, 3.2 and 3.3 provide a detailed confusion matrix for anomaly detection results using Dictionary Learning with LSA , TAD and our proposed method. We have also added the false positive rate (FPR) and Recall for each test.

Table 3.4 provide the summary by providing the mean FPR and Recall rate.

3.6 Conclusion

In this chapter we discuss a semi-supervised as well as an unsupervised technique to detect anomalies on time series data. Dictionary learning with Latent Semantic analysis given a very good recall rate but also has a high False Positive Rate which is undesirable. We also have a similar case with Topological Anomaly Detection. We observe that our algorithms are highly sensitive to the very large valued anomalies but comparatively less sensitive to anomalies smaller than the median value.

Table 3.4 summarizes our results comparatively. We observe that we achieve a better false positive rate than either of the discussed algorithm used individually. The next chapter provides information on future work that can be done.

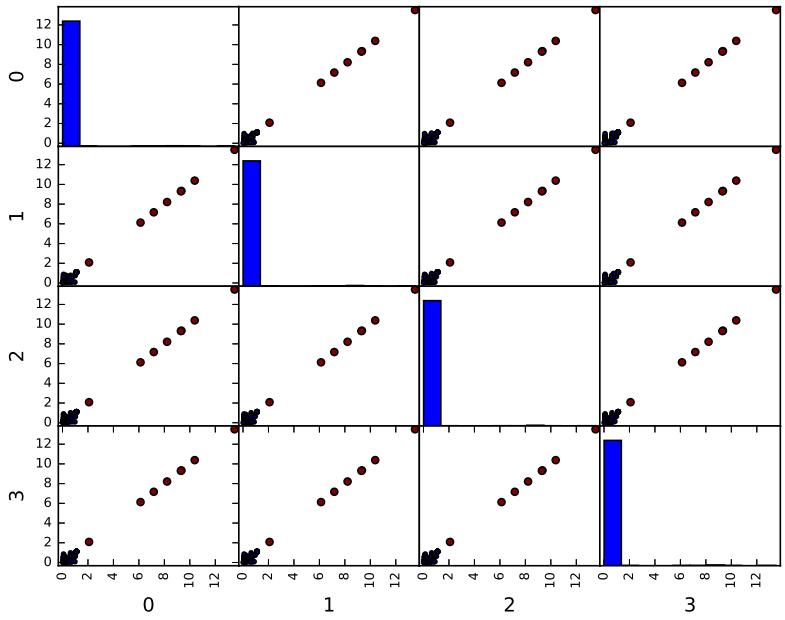


Figure 3.3: Pairs plot showing anomalies in House Number 1001

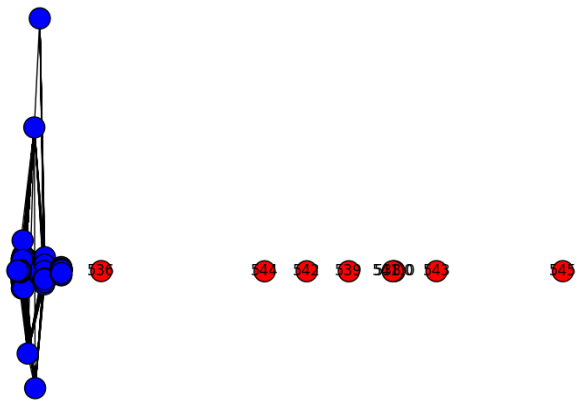


Figure 3.4: Visualization of Anomalies in house 1001

House	TA	TN	FP	FN	TP	FPR	Recall
1001	10	535	0	2	8	0.000000	0.8
1002	10	527	8	3	7	0.014953	0.7
1003	10	531	4	2	8	0.007477	0.8
1004	10	535	0	3	7	0.000000	0.7
1005	10	535	0	2	8	0.000000	0.8
1006	10	535	0	3	7	0.000000	0.7
1007	10	533	2	2	8	0.003738	0.8
1008	10	529	6	2	8	0.011215	0.8
1009	10	535	0	2	8	0.000000	0.8
1010	10	535	0	2	8	0.000000	0.8
1050	10	535	0	3	7	0.000000	0.7
1051	10	534	1	3	7	0.001869	0.7
1052	10	529	6	2	8	0.011215	0.8
1053	10	535	0	3	7	0.000000	0.7
1054	10	533	2	2	8	0.003738	0.8
1055	10	533	2	2	8	0.003738	0.8
1056	10	533	0	9	1	0.000000	0.1
1057	10	535	0	3	7	0.000000	0.7
1058	10	522	13	1	9	0.024299	0.9
1059	10	534	1	2	8	0.001869	0.8

Table 3.1: Topological Anomaly Detection

House	TA	TN	FP	FN	TP	FPR	Recall
1001	10	530	5	0	10	0.009434	1
1002	10	530	5	0	10	0.009434	1
1003	10	530	5	0	10	0.009434	1
1004	10	530	5	0	10	0.009434	1
1005	10	530	5	0	10	0.009434	1
1006	10	530	5	0	10	0.009434	1
1007	10	533	2	2	8	0.013258	0.8
1008	10	529	6	1	9	0.011342	0.9
1009	10	529	6	1	9	0.011342	0.9
1010	10	528	7	1	9	0.013258	0.9
1050	10	527	8	3	7	0.015209	0.7
1051	10	526	7	0	10	0.013258	1
1052	10	528	8	0	10	0.015180	1
1053	10	527	6	0	10	0.011342	1
1054	10	529	6	0	10	0.011342	1
1055	10	529	7	2	8	0.013258	0.8
1056	10	528	6	0	10	0.011342	1
1057	10	529	6	0	10	0.011342	1
1058	10	529	7	0	9	0.013258	0.9
1059	10	528	7	0	10	0.013258	1

Table 3.2: Dictionary Learning with Latent Semantic Analysis

House	TA	TN	FP	FN	TP	FPR	Recall
1001	10	535	0	2	8	0.000000	0.8
1002	10	534	1	3	7	0.001869	0.7
1003	10	531	4	3	7	0.007477	0.7
1004	10	535	0	1	9	0.000000	0.9
1005	10	535	0	3	7	0.000000	0.7
1006	10	534	1	3	7	0.001869	0.7
1007	10	533	2	3	7	0.003738	0.7
1008	10	530	5	4	6	0.009346	0.6
1009	10	535	0	2	8	0.000000	0.8
1010	10	535	0	2	8	0.000000	0.8
1050	10	535	0	3	7	0.000000	0.7
1051	10	535	1	3	7	0.001869	0.7
1052	10	533	2	1	9	0.003738	0.9
1053	10	535	0	3	7	0.000000	0.7
1054	10	534	1	2	8	0.001869	0.8
1055	10	533	2	1	9	0.003738	0.9
1056	10	535	0	2	8	0.000000	0.8
1057	10	535	0	3	7	0.000000	0.7
1058	10	535	0	1	9	0.000000	0.9
1059	10	534	1	2	8	0.001869	0.8

Table 3.3: TAD + DL

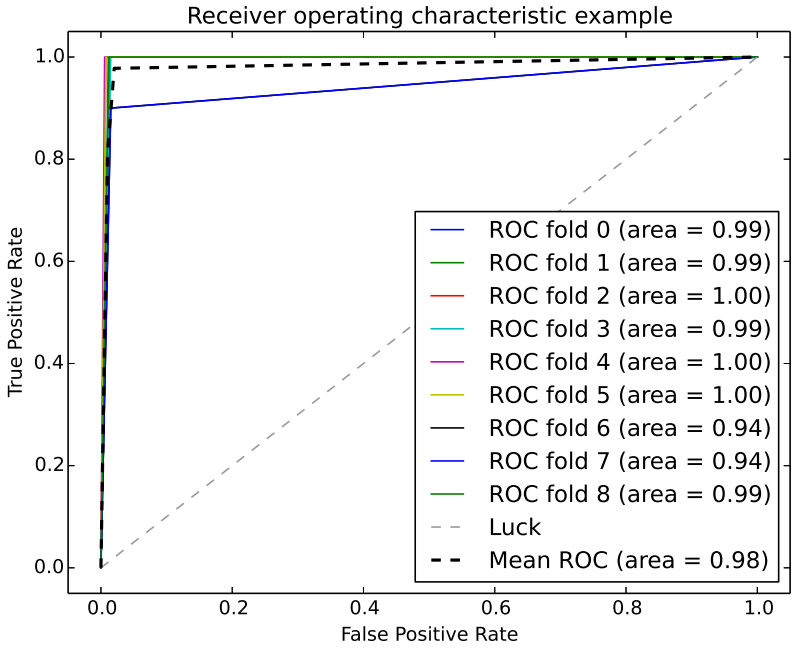


Figure 3.5: ROC for DL with LSA

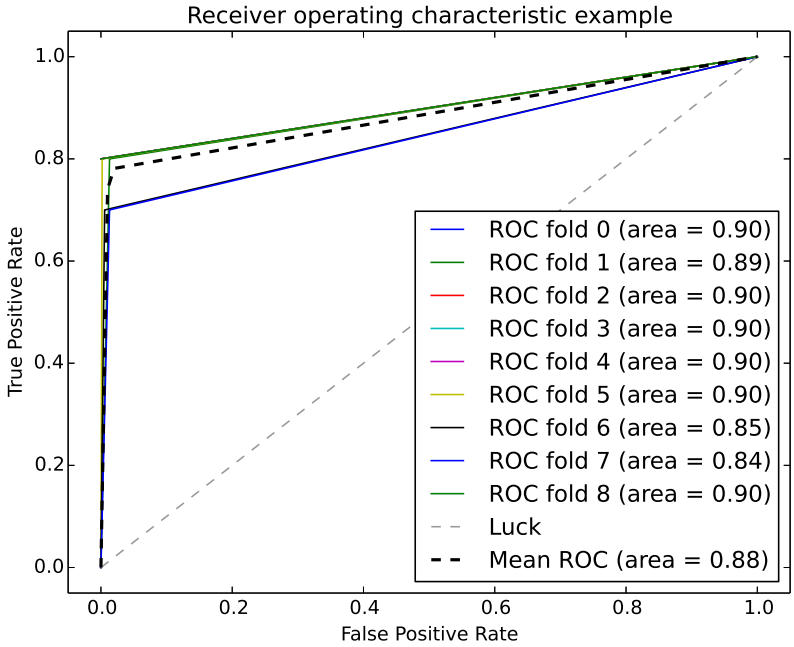


Figure 3.6: ROC for TAD

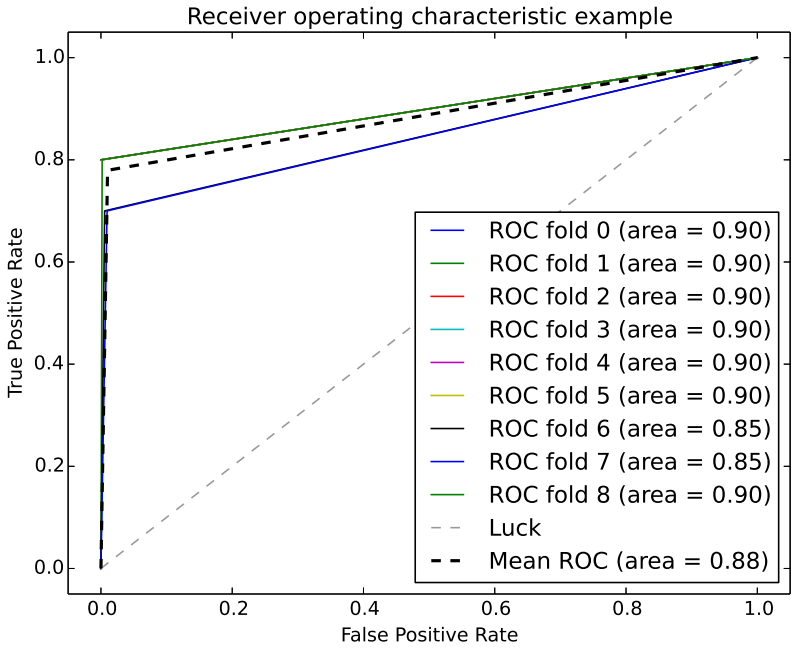


Figure 3.7: ROC for our proposed method

Parameter	Proposed Method	TAD	DL with LSA
Average FPR	0.1869%	0.4206%	1.173%
Average Recall	76.5%	73.5%	94.5%

Table 3.4: Result Summary

Chapter 4

Conclusion and Future Work

4.1 Summary of Contributions of the Thesis

The problem of anomaly detection on time series and discrete sequences can be divided into three broad categories. In Chapter 2 we have discussed all the three problem formulations in detail and the existing methods to solve them are presented. We have also presented the pros and cons of each presented method in the same chapter.

In Chapter 3 we use a semi-supervised method i.e Dictionary Learning combined with LSA and assume that approximately 1.5 percent of the points are anomalous in the data, we get a high recall rate but at the same time we also get a high false positive rate. When we used an unsupervised method i.e Topological Anomaly Detection it is quite a challenging task to set the tuning parameters.

Our proposed method gives an improvement of 91.4 and 75.2 percent in terms of False Positive Rate as compared to the semi-supervised method and unsupervised method individually.

Our proposed method loses 18 percent in terms of Recall Rate as compared to the semi-supervised method but gains 3 percent when compared to the unsupervised method individually.

As decreasing the FPR was our major concern we have achieved it by combining two separate methods and optimizing it according to the data set. We have successfully detected anomalies pertaining to tampering with smart meters with a 76.5 percent accuracy.

4.2 Future Work

On the basis of the fact that by combining two very different approaches to detection of anomalies on time series we are able to improve upon the FPR, creating an ensemble of such methods with more filters can improve the results even more. Implementation of the proposed method with larger dataset can refine the results even more.

Replacement of TAD with another semi-supervised method like Hidden Markov

Models can improve the Recall rate while keeping the false positive rate low. However this would still require reliability of the given dataset.

TAD and Sparse Coding have very high computational complexity and do not scale very well. Future work can include decreasing of the processing time by approximations of the available data.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 5, pp. 823–839, 2012.

[2] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection." in *SDM*. SIAM, 2003, pp. 25–36.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[4] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[5] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[6] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[7] S. Budalakoti, A. N. Srivastava, and M. E. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 101–113, 2009.

[8] J. Yang and W. Wang, "Cluseq: Efficient and effective sequence clustering," in *Data Engineering, 2003. Proceedings. 19th International Conference on*. IEEE, 2003, pp. 101–112.

[9] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*. IEEE, 1996, pp. 120–128.

[10] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[12] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[13] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 228–247, 2008.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[15] N. Takeishi and T. Yairi, "Anomaly detection from multivariate time-series with sparse representation," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on.* IEEE, 2014, pp. 2651–2656.

[16] M. Gartley *et al.*, "Topological anomaly detection performance with multispectral polarimetric imagery," in *SPIE Defense, Security, and Sensing.* International Society for Optics and Photonics, 2009, pp. 73 341O–73 341O.

[17] L. R. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[18] P. Jokar, N. Arianpoo, and V. Leung, "Electricity theft detection in ami using customers' consumption patterns," *Smart Grid, IEEE Transactions on*, vol. 7, no. 1, pp. 216–226, 2016.

[19] M. Elad, "Sparse and redundant representation modeling—what next?" *Signal Processing Letters, IEEE*, vol. 19, no. 12, pp. 922–928, 2012.

[20] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.