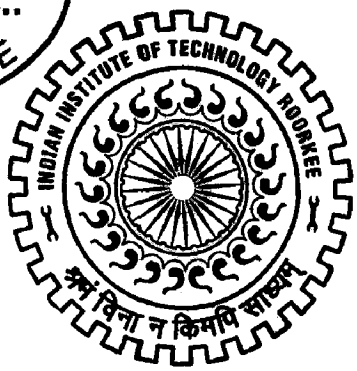
ADAPTIVE OUTLIER DETECTION IN STREAMING TIME SERIES

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By

SHACHI YADAV



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)

JUNE, 2010

CANDIDATE'S DECLARATION

I hereby declare that the work being presented in the dissertation report entitled **"Adaptive Outlier Detection in Streaming Time Series"** in partial fulfilment of the requirement for the award of the degree of Master of Technology in Computer Science and Engineering, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee, is an authentic record of my own work carried out under the guidance of Dr. Durga Toshniwal in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee. The matter embodied in the dissertation report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

Dated: 29/06/10

Place: I.I.T Roorkee

Shachi Yadav

Shachi Yadav

CERTIFICATE

This is to certify that above statements made by the candidate are correct to the best of our knowledge and belief.

Dated:

Place: I.I.T Roorkee

Durga Toshniwal 29/6/10.

Dr. Durga Toshniwal

Assistant Professor,

Department of Electronics and Computer Engineering

ACKNOWLEDGEMENTS

First of all and foremost, I would like to express my deep sense of gratitude and indebtedness to my guide Dr. Durga Toshniwal, for her invaluable guidance and constant encouragement throughout the dissertation. I would like to sincerely thank her for allocating me resource in Computer Centre for the purpose of this work. I am also thankful to each and every member of the research scholar laboratory for the constant encouragement and cooperation.

My special sincere heartfelt gratitude to my family, whose best wishes, support and encouragement has been a constant source of strength to me during the entire work. Finally, I would also like to thank all my friends and seniors for their support and valuable suggestions.

Shachi Yadav

Abstract

"Outlier" is a scientific term to describe things or phenomena that lie outside normal expectation or behaviour. In data mining outlier detection is a type of data analysis technique that seeks to determine and report such data objects which are grossly different from or inconsistent with the remaining set of data. The technique is used for data cleansing, spotting emerging trends and recognizing unusually good or bad performers. Typical applications are financial data analysis, intrusion detection, event detection in sensor networks, biomedicine etc.

The existing outlier detection schemes aim to detect the global outliers from the entire time series data and therefore fail to detect the local outliers. The detection of local outliers is helpful as they tell the degree of isolation of objects from their immediate neighbourhood. The existing schemes process outliers by working on the entire outlier time sequences. But in case of streaming time series data, this is not possible as the data keeps on arriving from the source.

In the proposed work, we aim to develop an algorithm that detects outliers from streaming time series. The outliers are extracted as abnormally behaving subsequences in the data. The emphasis is on detecting the local outliers in addition to global outliers. The notion of "outlierness" has also been introduced which is used to capture the extent of abnormal behaviour shown by the outliers. Further, the type is also defined. It refers to the deviation of outliers above or below the normal behaviour. The HOT SAX algorithm has been extended to detect the local outlier subsequences in the time series streams. The outlier distribution is generated on the basis of reference set, to develop a rule based adaptive model to classify outliers into local and global classes.

The proposed work has been evaluated on real life datasets. The first dataset used is a daily vehicular traffic dataset, that is, Gotthard tunnel dataset- number of motorcycles in one direction (in year 2005). The other dataset used is ECG dataset.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1Data Mining

Across a wide variety of fields, data are being collected and accumulated at a dramatic pace. There is an urgent need for a new generation of computational theories and tools to assist humans in extracting useful information from the rapidly growing volumes of digital data [1].

Of late, data mining in databases have been attracting a significant amount of interest in wide variety of areas, such as science, marketing, finance, health care, retail, and many other fields [1]. By performing data mining the discovered knowledge can be applied to decision making, process control, information management, and query processing [1]. Some widely used data mining methods are [2]: concept/class description (that is characterization and discrimination), mining frequent patterns, associations and correlations, classification and prediction, cluster analysis, and outlier analysis.

1.2 Time Series Streams

The past decade has seen a wealth of research on time series data. The vast majority of research has concentrated on data mining techniques that are calculated in batch mode and can store in physical media [3].

However, data streams have received considerable attention in various communities due to the increasing deployment of mobile devices and real time sensors such as in network analysis, sensor network monitoring, moving object tracking, financial data analysis, and scientific data processing. All these applications have in common that, (i) massive amounts of data arrive at high rates, which make traditional database systems prohibitively slow, and (ii) users, or higher-level applications, require immediate responses with high accuracy [3]. These two requirements have brought home the need for mining techniques that is adaptable with the continuous inflow of data, and can provide with optimum storage requirement.

1.3 Outlier Analysis

Outliers are the data objects that do not comply with the general behaviour or model of the data [2]. Figure 1.2 shows two data objects as outliers O_1 and O_2, isolated from other data objects in C_1 and C_2. Searching for outliers in data streams is an important area of research in the world of data mining with numerous applications, including credit card fraud detection, discovery of criminal activities in electronic commerce, weather prediction, etc [4].

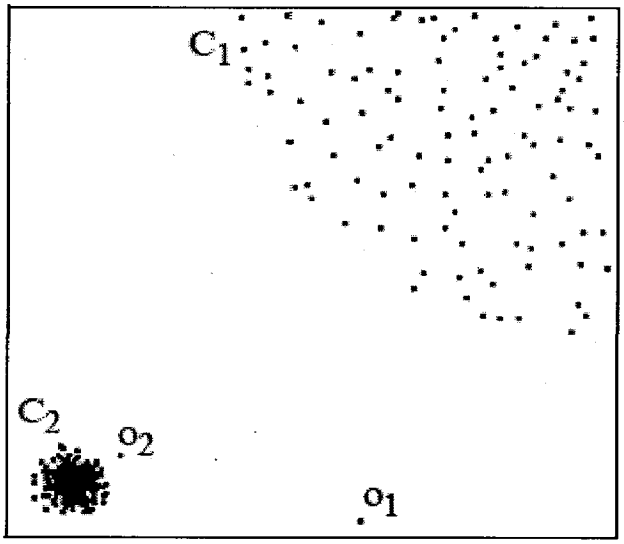


Figure 1.1: Shows two isolated data objects as outliers O_1 and O_2

Recently mining outliers in data stream attracts more and more attention. It becomes a challenge work due to the characteristics of data streams. Stream data differ from conventional stored relational data since a data stream is an ordered sequence of data that arrive continuously and change fast [4]. It is unlimited in size and not possible to save in a physical media. Thus it needs to be processed differently to quickly extract nearly real-time information.

1.4 Motivation

Recently outlier detection in stream data is becoming very important, as discussed in section 1.3, due to its wide application in areas such as Internet traffic analysis, sensor network monitoring, moving object search, financial data analysis, and the like. Most existing outlier detection techniques, however, process outliers by working on the

entire outlier time sequences, which is computationally expensive and incapable to adapt with the continuously incoming data streams. Also, these schemes aim to detect the global outliers from the entire time series data and therefore fail to detect the local outliers. The detection of local outliers is helpful as they tell the degree of isolation of objects from their immediate neighbourhood.

1.5 Problem Statement

The objective of this dissertation work is as follows:

"Adaptive detection of outlier subsequences from streaming time series data".

The following sub-problems have been addressed:

- To extend the Heuristically Ordered Time series using Symbolic Aggregate Approximation (HOT SAX) algorithm to extract local outlier subsequences from data streams.
- To develop an adaptive rule-based classification model for classifying outliers into local or global classes.
- To establish the type for the outliers. That is, to further classify the outliers as "above normal" or a "below normal" depending upon the deviation shown by them about the normal behaviour.
- To identify the degree of "outlierness". That is, to capture the extent of abnormal behaviour shown by outliers in terms of severe or mild abnormal behaviour.

1.6 Organization of the Report

The organization of this dissertation report is as follows:

Chapter 2 gives a brief overview of the distance measure and the classification method. The literature review is also done of various outlier techniques been proposed so far, for time series data. The research gaps found in them are also discussed with a possible solution for it, which we proposed in our work.

Chapter 3 discusses the proposed work. It provides the overall architecture for the proposed algorithm. The various subsections provide the details of the proposed algorithm for detecting the adaptive outliers in time series stream data.

Chapter 4 presents the description of datasets used, results of the experiments performed and its analysis.

Chapter 5 concludes the report and also presents the suggestions for future work.

Chapter 2: Literature Review

2.1 Distance Metric

A distance metric measures the dissimilarity between two data points in terms of some numerical value. It also measures similarity; we can say that more distance less similar and less distance more similar [2].

To define a distance metric, we need to designate a set of points, and give a rule, $d(X, Y)$, for measuring distance between any two points, X and Y, of the space. Mathematically, a distance metric is a function, d, which maps any two points, X and Y in the n-dimensional space, into a real number, such that it satisfies the following three criteria [2].

Criteria of a Distance Metric

- ***d(X, Y) is positive definite:*** If the points X and Y are different, the distance between them must be positive. If the points are the same, then the distance must be zero. That is, for any two points X and Y,

  i. if $(X \neq Y)$, $d(X, Y) > 0$

  ii. if $(X = Y)$, $d(X, Y) = 0$

- ***d(X, Y) is symmetric*:** The distance from X to Y is the same as the distance from Y to X. That is, for any two points X and Y,

  $$d(X, Y) = d(Y, X)$$

- ***d(X, Y) satisfies triangle inequality*:** The distance between two points can never be more than the sum of their distances from some third point. That is, for any three points X, Y and Z,

  $$d(X, Y) + d(Y, Z) \geq d(X, Z)$$

Among the possibilities, Manhattan, Euclidian, and Max distance metrics are common.

By far the most common distance measure for time series is the Euclidean distance measure [5]. Given two time series Q and C of the same length n, their Euclidean distance is measured by the following equation (2.1):

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2} \tag{2.1}$$

It is simple to understand and easy to compute. It also allows scalable solutions for the problems such as time series indexing and clustering.

2.3 Streaming Time Series Classification

Classification is a form of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with a better understanding of the data at large.

Data classification is a two-step process [2]:

- **First step:** a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analysing or "learning from" a training set made up of database tuples and their associated class labels.
- **Second step:** the model is used for classification. First, the predictive accuracy of the classifier is estimated a test set is used, made up of test tuples and their associated class labels. These tuples are randomly selected from the general data set. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.

The above discussed traditional classification process is unable to deal with the streaming time series data. The major difficulties introduced in applying the traditional classification methods to streaming time series are discussed below [2]:

- The traditional techniques will scan the training data multiple times. The first step of model construction is typically performed off-line as a batch

process. In streaming time series the data flow in so quickly that storage and multiple scans are infeasible.

- In traditional schemes decision tree algorithms tend to follow the same basic top-down, recursive strategy, yet differ in the statistical measure used to choose an optimal splitting attribute. However, in the stream environment, it is neither possible to collect the complete set of data nor realistic to rescan the data.
- The most distinguishing characteristic of data streams is that they are time-varying, as opposed to traditional database systems, where only the current state is stored. This change in the nature of the data takes the form of changes in the target classification model over time and is referred to as concept drift. Concept drift is an important consideration when dealing with stream data.

2.4 Outlier Detection Techniques

The existing outlier detection methods that have been proposed so far over static data and stream data are discussed below:

Statistical Based Approaches

These methods assume that the dataset follows a statical model, e.g. a normal or Poisson distribution. With these methods we detect objects that deviate from the model as outliers using a *discordancy test*. Application of the test requires knowledge of the data set parameters such as the assumed data distribution, knowledge of distribution parameters such as the mean and variance, and the expected number of outliers [2].

There are two basic types of procedures for detecting outliers in this approach that is [2]:

- *Block procedures:* In this case, either all of the suspect objects are treated as outliers or all of them are accepted as consistent.
- *Consecutive (or sequential) procedure:* An example of such a procedure is the *insideout* procedure. Its main idea is that the object that is least "likely" to be an outlier is tested first. If it is found to be an

outlier, then all of the more extreme values are also considered outliers; otherwise, the next most extreme object is tested, and so on. This procedure tends to be more effective than block procedures.

However, statistical approaches have following major drawbacks [2, 6]: these approaches make a lot of assumptions about the distribution model, and have difficulty dealing with streams, and do not guarantee that all outliers will be found.

Cluster Based Approaches

The cluster based approaches, such as CLARANS [7], DBSCAN [8], BIRCH [9], Wave Cluster [10], CLIQUE [11], etc have been used for outlier detection in diverse datasets. The main problem of these clustering approaches is that they detect outliers as by-products. In most cases, the main objective is to find clusters in the dataset. For that reason, this approach sometimes does not focus entirely on outlier detection [12].

Density-Based Approaches

These methods adopt a Local Outlier Factor (LOF) for outlier detection [13]. It assigns to each object an outlier factor with respect to its surrounding neighbourhood. The outlier factor depends on how the data object is closely packed in its locally reachable neighbourhood. Since LOF uses threshold to differentiate outliers from normal objects the same problem of parameter setting arises. A lower outlier-ness threshold will produce problem of false detection rate, while a high threshold value will result in missing genuine outliers [14]. Besides top-*n* and top-*n* LOF, other approaches are connectivity–based (COF) [15] and Resolution cluster-based (RB-outlier) [16] they alleviate the difficulty of parameter setting but their detection method is not generic in nature.

Distance-Based Approaches

The distance based approach is a simpler and more common approach. Several efficient algorithms for mining distance-based outliers have been developed. Such as index based algorithms, nested loops or cell based [2]. The Local Distance–based Outlier Factor (LDOF) [14] approach is used for outlier detection in scattered data. It simply uses the relative location of an object to its neighbours to determine the degree to which the object deviates from its neighbourhood. The Continuous Distance

Based-Outlier detection (CDB-Outliers) [12] approach is used for continuously detecting outliers over stream data. It employs DB-Outlier, based on the Cell Based algorithm for quick processing.

2.5 Research Gaps

Some of the research gaps found in the previous work are:

- Statistical approaches make a lot of assumptions about the distribution model, have difficulty dealing with stream data, and do not guarantee that all outliers will be found.
- The main problem of clustering approaches is they detect outliers as by-products. In most cases, the main objective is to find clusters in the dataset. For that reason, this method does not focus entirely on outlier detection.
- Density based approaches mainly suffer from the requirement of setting parameter values correctly.

Hence, the proposed work undertakes to present a methodology that work on the above mentioned research gaps by following these strategies:

- By developing an approach that adaptively builds a distribution model so as to capture the real nature and characteristics of the data and does not have to make unnecessary assumptions.
- By utilising the concept of detecting outliers in local segments, and then classifying them into local or global outliers will alleviate the problem of limited detection of outliers.
- By extending the HOT SAX algorithm, that requires only one parameter and that too the length of the outlier subsequence, in streaming time series, the proposed approach does not suffer from the difficulty of setting too many parameter values, correctly.

Chapter 3: Proposed Work

3.1 Overall Architecture

The overall architecture for proposed algorithm is presented in the following Figure 3.1. The block diagram provides an overview of flow of control between various blocks of the proposed algorithm. It also shows the function performed by each block.

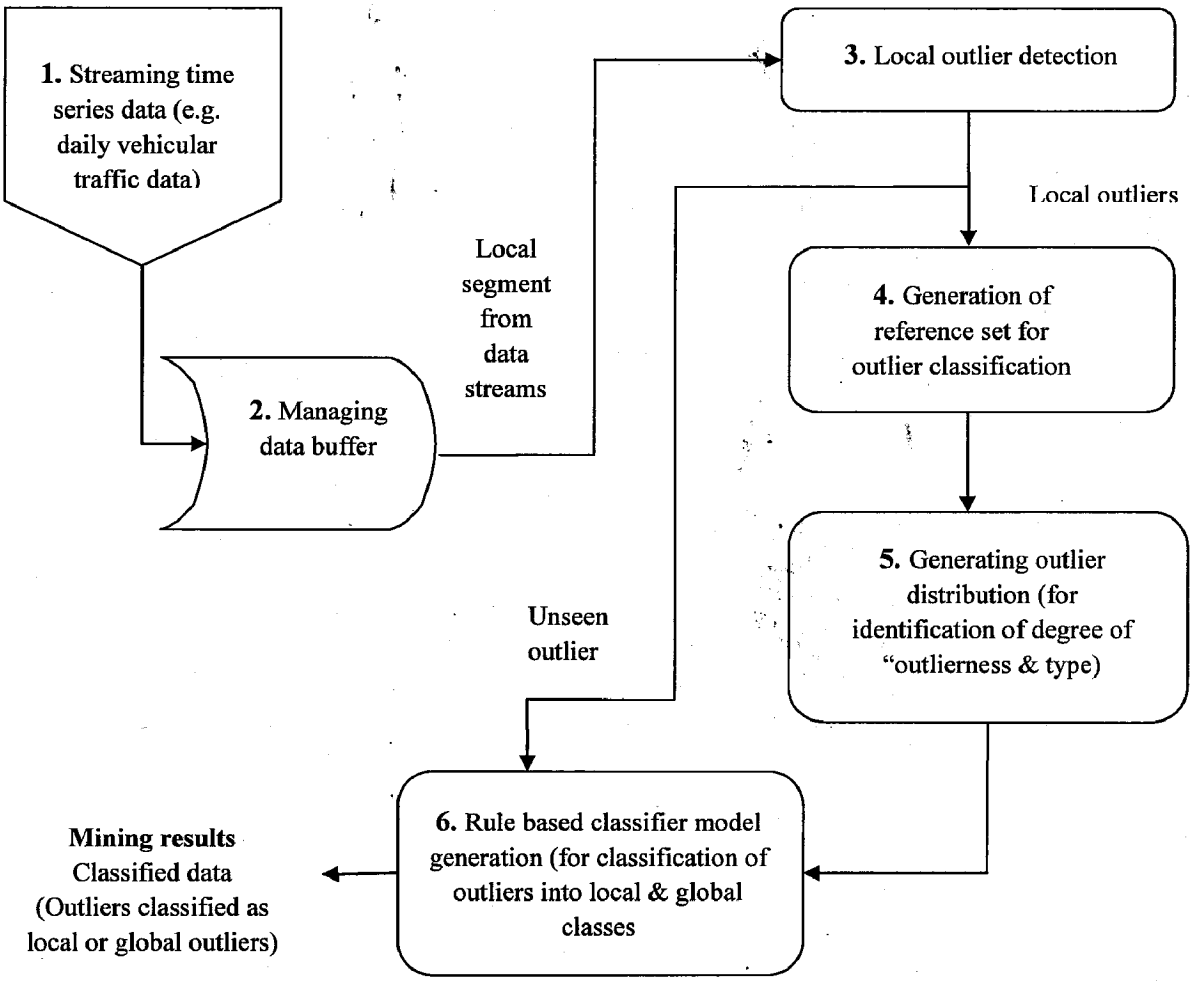


Figure 3.1 The overall architecture for proposed algorithm

A brief description of the block diagram of Figure 3.1 is as follows:

The time series stream data is being generated by some source, say by monitoring of vehicular traffic on a highway, is captured and stored in data buffer. The capacity of data buffer is limited and predetermined, and is dependent on the data that is being used. The sliding window concept is used to manage the storage of data in the buffer. That is, as new data objects are generated, they are inserted into the beginning of the buffer and a corresponding number of data objects are removed from the end of the buffer.

In each local segment, collected in data buffer from stream data, the local outlier is detected. The HOT SAX algorithm has been extended to find the local outliers in local streams. The Local outliers detected are stored in a vector of predetermined capacity. The capacity of this vector is also dependent on the data that is being used and, follows the sliding window concept to manage the storage of outliers. The vector is used for generating reference set. Once this vector is full to its maximum capacity, the first reference set is generated.

The reference set is used for generating the outlier distribution. It is generated by measuring the dispersion of local outliers received in the reference set. Then rules are formulated which are used for classification of outliers and further, for identifying the degree of "outlierness" and the type of these outliers.

Rule-based classification model is used to classify the outliers into local or global classes. This model is adaptive in the sense that, every time the reference set is updated by new arrival of local outliers, the outlier distribution is also updated. The rules formulated on the outlier distribution are also updated. Every time the rules are updated, the classification model adapts itself according to the dynamics of streaming time series. Thus, we are able to capture the adaptive outliers in the streaming time series.

The following sections provide a detailed description of the functions performed by each block shown in the Figure 3.1.

3.2 Data Buffer

The data buffer is used to store the captured time series stream data generated by some source, say by monitoring of vehicular traffic on a highway. The capacity of data buffer is limited and predetermined, and is dependent on the data that is being used. The sliding window concept is used to manage the storage of data in the buffer. The concept of sliding window is defined as follows:

Sliding window: Suppose, we use time series data buffer say ***DB*** to contain the local segment of flowing time series. In general, at a certain time point t, suppose time series buffer ***DB*** contains a time series of length n, that is, $DB \equiv T = t_1, t_2, ..., t_n$, and t_{new} is the next arriving time series data point. At next time $t+1$, the time series buffer data is changed as $DB \equiv T' = t_2, ..., t_n, t_{new}$. The Figure 3.2 clearly explains the sliding window concept [2].

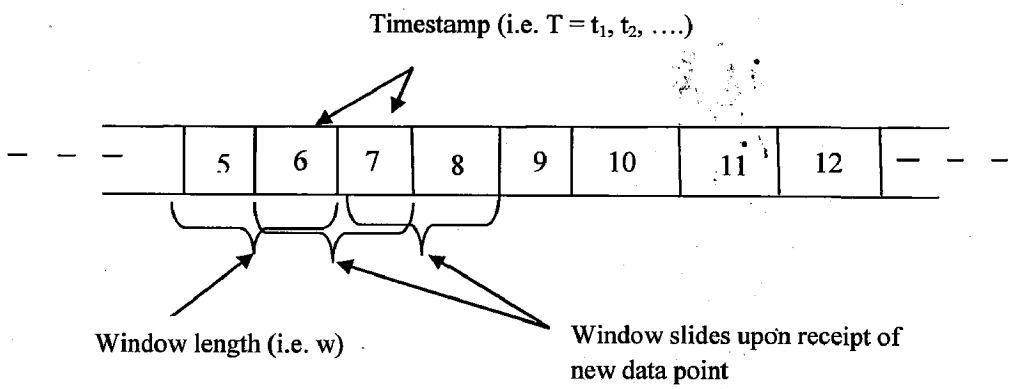


Figure 3.2: Explains the concept of sliding window for streaming time series.

We use the notation ***(m, n) (t)*** to denote the time series subsequence in the data buffer. Where m is the position in the data buffer and n is the length of the subsequence at time t. Therefore, ***(p-1, n) (t+1)*** and ***(p, n) (t)*** are the same time series subsequence for a given time series stream, at two different timestamps t and $t+1$.

3.3 Local Outlier Detection

The HOT SAX algorithm [17] has been extended to find the local outliers in each local segment, collected in data buffer captured from streaming time series data. It follows a simple approach for finding local outliers in the given local segment of streaming time series data. It simply takes each possible subsequence and finds the distance to the nearest non-self match. The subsequence that has the greatest such value is the outlier. This is achieved with nested loops, where the outer loop considers each possible candidate subsequence, and the inner loop is a linear scan to identify the candidate's nearest non-self match.

Before presenting the details of HOT SAX algorithm some basic notations and concepts that have been used in it are being discussed as follows [17]:

- **Non-Self Match:** Given a time series T, containing a subsequence C of length n beginning at position p and a matching subsequence M beginning at q, if $|p - q| \geq \mathbf{n}$, we say M is a non-self match of C and their distance is $Dist(M, C)$.
- **Non-Similar Distance**: Given a time series T, for any subsequence P of T, Q is the nearest non-self match of P, the distance from P to Q is the non-similar distance of P.
- **Time Series Outlier:** Given a time series T, the subsequence D of length n beginning at position l is said to be the outlier of T if D has the largest distance to its nearest non-self match. That is, $\forall$ subsequences C of T, non-self matches MD of D, and non-self matches MC of C, $\min(Dist(D, MD)) > \min(Dist(C, MC))$.

- **Symbolic Aggregate Approximation (SAX) [5]:** allows a time series of arbitrary length n to be reduced to a string of arbitrary length w, ($w < n$, typically $w << n$). It first transforms the data into the Piecewise Aggregate Approximation (PAA) representation [5]. Then symbolizes the PAA representation into a discrete string, two parameters are required for it. The SAX word size w is the number of symbols required to represent the

subsequences in the low dimensional approximation, and the cardinality of the SAX alphabet size α, that is the number of discrete symbols needed to represent the SAX word.

Brief Review of HOT SAX Algorithm

The HOT SAX algorithm follows the following procedure to find the outlier in the given time series data [17]:

- **SAX representation of the time series:** A SAX representation of the given time series is created first, by sliding a window of length say n across the given time series. The length of the outlier subsequence n is given in advance.

- **Creation of two data structures:** HOT SAX algorithm creates two data structures to support the nested loop heuristics. The two heuristics are used for optimization of nested loop computation. The first data structure is an array containing extracted subsequences converted into SAX words, where the index refers back to the original sequence. Once we have this ordered list of SAX words, we can imbed them into an augmented trie another data structure where the leaf nodes contain a linked list index of all word occurrences that map there. The count of the number of occurrences of each word can be mapped back to the rightmost column of the array.

- **Find the distance to the nearest non-self match:** The nested loops are used for this purpose, where the outer loop considers each possible candidate subsequence, and the inner loop is a linear scan to identify the candidate's nearest non-self match. To efficiently find the distance to the nearest non-self match the nested loop heuristics are used. The subsequence that has the greatest such value is declared as outlier.

The intuition behind the two heuristics is as follows [17]; unusual subsequences are very likely to map to unique or rare SAX words. By considering the candidate sequences that map to unique or rare SAX words early in the outer loop, there is an excellent chance of giving a large value to a variable used to store the best distance found so far, early on, thus allowing more early terminations of the inner loop. The

inner heuristic also leverages off the two data structures. When candidate say ***i*** is first considered in the outer loop, we look up the SAX word that it maps to, by examining the *ith* word in the array. We then visit the trie and order the first items in the inner loop in the order of the elements in the linked list index found at the terminal nodes. The Figure 3.3 shown below gives the visual intuition of the two data structures that are being used to support the two heuristics.

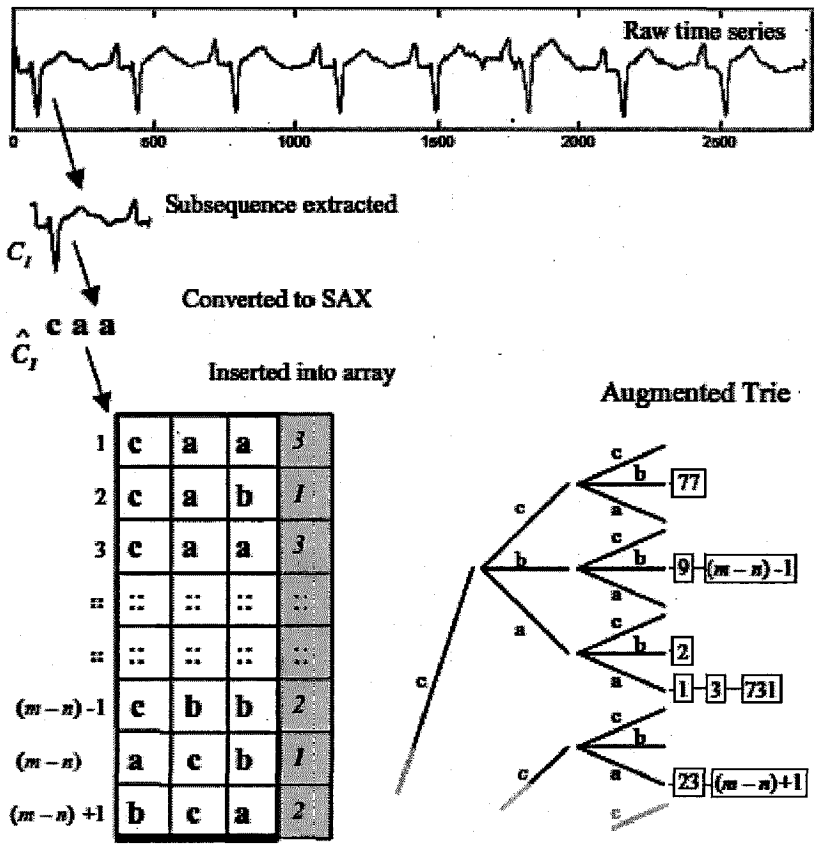


Figure 3.3: The two data structures used to support the *Inner* and *Outer* heuristics in HOT SAX algorithm.

3.4 Reference Set Generation

Since we are dealing with streaming time series data we don't have any prior data knowledge so as to build a classifier for describing a predetermined set of data classes or concept for learning, as used in traditional classification method. Therefore, in the proposed algorithm we generate a reference set that can be used for learning the

nature of stream data that is being used. This reference set is continuously updated so as to capture the dynamics of the streaming time series.

The Local outliers detected from local segments are stored in a vector of predetermined capacity. The capacity of this vector is dependent on the data that is being used and, follows the sliding window concept to manage the storage of outliers. The vector is used for generating the reference set. The reference set is used for generation of outlier distribution. Once this vector is full to its maximum capacity, the first reference set is generated.

3.5 Outlier Distribution Generation

The outlier distribution is generated for formulating rules for classification model. It is generated by measuring the dispersion of local outliers received in the reference set. The measurement of dispersion of data is very useful in providing typical properties and characteristics about the data [2]. The most common measures of data dispersion are quartiles, interquartile range, standard deviation, etc.

In the proposed algorithm we calculate the first quartile, the third quartile and the interquartile range to measure the dispersion of local outliers stored the in reference set.

The definition of these three measures of dispersion is as follows [2]:

- **The first quartile:** Also dented by Q_1, is the 25^{th} percentile of a set of data sorted in increasing order. This means that 25 percent of the data entries in a given sorted data set lie at or below the value of first quartile. The first quartile is calculated as the median of the lower half of the data, sorted in increasing order.
- **The third quartile:** Also dented by Q_3, is the 75^{th} percentile of a set of data sorted in increasing order. This means that 75 percent of the data entries in a given sorted data set lie at or below the value of third quartile. The third quartile is calculated as the median of the upper half of the data, sorted in increasing order.

- **Interquartile range (IQR):** It is defined as the distance between the first and third quartile. It is a simple measure of spread that gives the range covered by the middle half of the data.

Now we present how the three measures of dispersion are calculated from the reference set:

The non-similar distance of local outliers collected in reference set are first sorted in an increasing order. Then the first and the third quartiles are calculated by finding the median value of the upper and the lower half of the reference set. These two values are then used to calculate the interquartile range. The calculated interquartile range is stored to generate the conditions for the rule used for classification of outliers into global or local outliers.

Every time the reference set is updated the value of these three measures of dispersion is also re-evaluated. If the value of the new interquartile range is same as the previous value of the interquartile range then the conditions of the rule used for classification is not re-evaluated, otherwise the new values are set for the conditions of the rule. By doing this we are able to generate an adaptive classifier model that is capable of capture the dynamic nature of the streaming time series.

3.6 The Rule-Based Classifier Model

The rule-based classification model is used to classify the outliers into local or global classes. This model is adaptive in the sense that, every time the reference set is updated by new arrival of local outliers, the outlier distribution is also updated. Therefore, rules formulated on the outlier distribution are also updated accordingly. We have defined the three measures of dispersion in section 3.5 and now we will formulate the rules for classification model.

Now we present the conditions for rule formulation are developed as follows:

Condition 1: Identifying repeated outliers

To check this condition we compare the location, in the data buffer, of the newly detected local outlier, say at time $t+1$, with the location of the previously detected

outlier, at time t. If the location is different from the previous location we consider the newly detected outlier as distinct, else we consider it as repeated.

That is, if we say that local outlier at time t is $LO(l, n)\ (t)$ and the local outlier at time $t+1$ is $LO(k, n)\ (t+1)$, then they are distinct if $l \neq k$. As discussed in section 3.2 that, $(p-1, n)\ (t+1)$ and $(p, n)\ (t)$ are the same time series subsequence for a given time series stream, at two different timestamps t and $t+1$, therefore two local outliers detected at time t and $t+1$ are distinct if:

$C1$: (Value of location at time $t+1$) $\neq$ ((Value of location at time t) – 1)

The condition $C1$, is thus used to prevent the classification of repeated outliers. Now we will present the next condition that is used to determine the outliers that are interesting from the rest of local outliers detected so far.

Condition2: Avoiding trivial local outliers

The three measure of dispersion defined provide a holistic view of the data distribution. Based on this holistic view of data distribution, a common rule of thumb for identifying suspected outliers in a given set of data is to single out values falling at least **1.5 * *IQR*** above the third quartile or below the first quartile [2]. Therefore, to determine whether the newly detected local outlier is an interesting outlier or not, we have generated two conditions, which are given as follows:

$C2$: Unseen outlier < (1.5 * $IQR - Q_1$)

$C3$: Unseen outlier > (1.5 * $IQR + Q_3$)

The conditions C2 and C3 are thus used to determine whether the unseen outlier is interesting from the rest of the outliers detected so far.

The conditions generated above will now be used to formulate the classification rule for classifying the outliers into global and local. Therefore, the rule generated from the above three conditions is:

Rule: IF $C1 \wedge (C2 \vee C3)$ THEN *global outlier = yes*

Now we present how the classifier utilises the above rule to classify outliers into local or global and also to establish the type of the global outlier detected:

The rule generated first checks whether the unseen outlier is a repeated outlier or not, that is it checks the condition ***C1***. If the unseen outlier is a repeated one, it is discarded else it checks the other two conditions. The other two conditions are used to determine wether the unseen outlier is interesting or not. If any one of the two conditions is fulfilled the classifier declares the unseen local outlier as global outlier, otherwise it is a local outlier.

Identification of degree of "outlierness" and type for the global outliers

Based on these rules the proposed algorithm also establishes the type of the global outlier. If the non-similar distance of the unseen outlier fulfilled the condition ***C2*** then it is considered as "below normal" type of outlier. Otherwise, if the condition ***C3*** is fulfilled the outlier is an "above normal" type.

The degree of "outlierness" is also identified for both types of outliers. The proposed algorithm considers the length of reference set as the set of degrees that will be mapped to outliers. We first sort the two vectors containing "above normal" and "below normal" outliers in the decreasing order. In the "above normal" type the severe degree is mapped to that outlier that has the largest value of non-similar distance and then the rest are considered as having mild degree. Whereas in the "below normal" type the severe degree is mapped to that outlier that has the smallest value of the non similar distance, and rest of them are considered as mild.

Chapter 4: Results and Discussions

This dissertation work has been implemented in Matlab R2006a running in Windows Vista. All the experiments were performed and results are obtained on Intel Core 2 Duo 2.10 GHZ processor with a 4 GB RAM.

The proposed work has been evaluated on real life datasets. The first dataset used is a daily vehicular traffic dataset, that is, Gotthard tunnel dataset- number of motorcycles in one direction (in year 2005). The other dataset used is ECG dataset. We have used colour scheme to show the degree of "outlierness" to which the global outliers are mapped to. The red colour is used to show the "severe" degree and the green is used to depict the "mild" degree. The type of outlier can be visualised from the graphs, that is wether it is of "above normal" type or "below normal" type.

4.1 Gotthard Tunnel Dataset- Number of Motorcycles in One Direction (in Year 2005).

The **St. Gotthard Tunnel** in Switzerland is the third longest road tunnel in the world. This road forms part of the shortest road link from Hamburg, Germany to Sicily in Italy. The dataset consists of daily transportation data. It measures the frequency of motorcycles passing in one direction through Gotthard tunnel each day in a week. It contains 365 records from Jan 2005 to Dec 2005 [18].

We first present the analysis of general traffic conditions in Gotthard tunnel all the year round and then we analyze the Gotthard tunnel dataset- Number of motorcycles in one direction, that is being used by the proposed algorithm for the detection of outliers, to visualize the anomalies in the dataset.

Analysis of general traffic conditions in Gotthard tunnel all the year round

The Gotthard Road Tunnel is open all the year round. Normally the traffic is fluid, too. The peak days are Easter and also the beginning of the Italian Summer holidays that is from mid May to August. A lot of Italians who live in Switzerland then drive

back their home-country, meaning that the traffic really increases [21]. We have used two datasets of year 2002 and 2003 of Gotthard Tunnel to prove the above mentioned statements.

These two datasets consists of number of all types of vehicles passing in both direction per day through Gotthard tunnel in year 2002 and 2003 [18]. The Figure 4.1 and Figure 4.2 are used as proof of the above mentioned statements and Figure 4.3 provides the evidence of the above statements. Table 4.1 is used to explain the analysis shown in the Figure 4.1 and Figure 4.2.

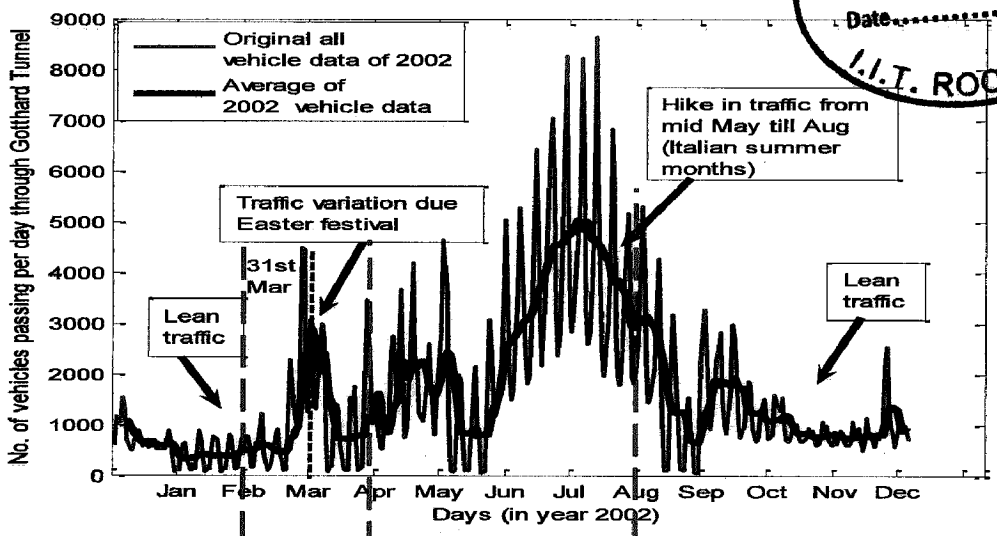


Figure 4.1: The Gotthard tunnel traffic of all types of vehicles in both direction in year 2002.

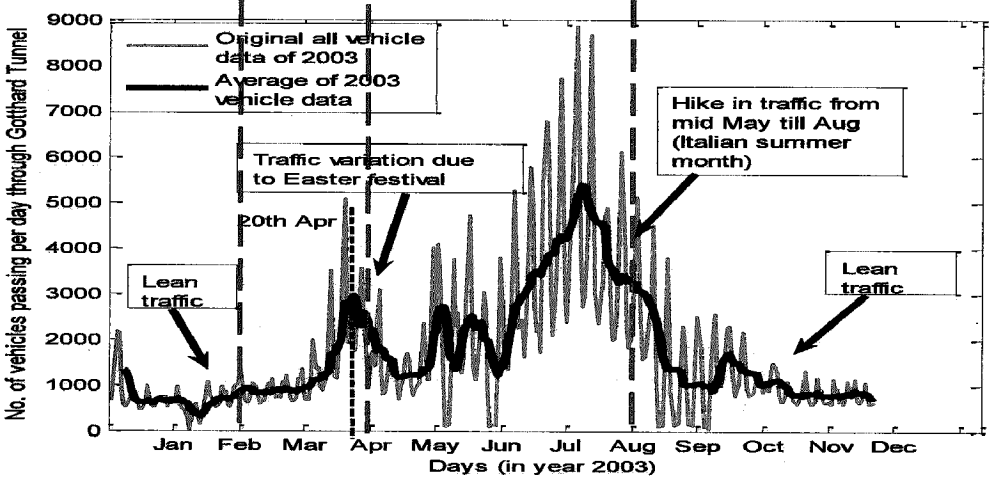


Figure 4.2: The Gotthard tunnel traffic of all types of vehicles in both direction in year 2003.

The analysis presented in the above two graphs is explained in the following Table 4.1:

Table 4.1: The explaination of the two graphs of Gotthard tunnel traffic in year 2002 and 2003 showing the general traffic all the year round.

Section of the graph explained	Traffic condition in year 2002	Traffic condition in year 2003
Section 1 (Jan to Feb)	Lean traffic	Lean traffic
Section 2 (Mar to Apr)	Traffic is high in last week of March due to Easter falling on 31st of March'02	Traffic is high in last week of April due to Easter falling on 20th ofApril'03
Section 3 (May to Aug)	Hike in traffic	Hile in traffic
Section 4(Sept to Dec)	Lean traffic	Lean traffic

Evidence: The evidence is presented here of the above analysis describing the Gotthard tunnel traffic all the year round.

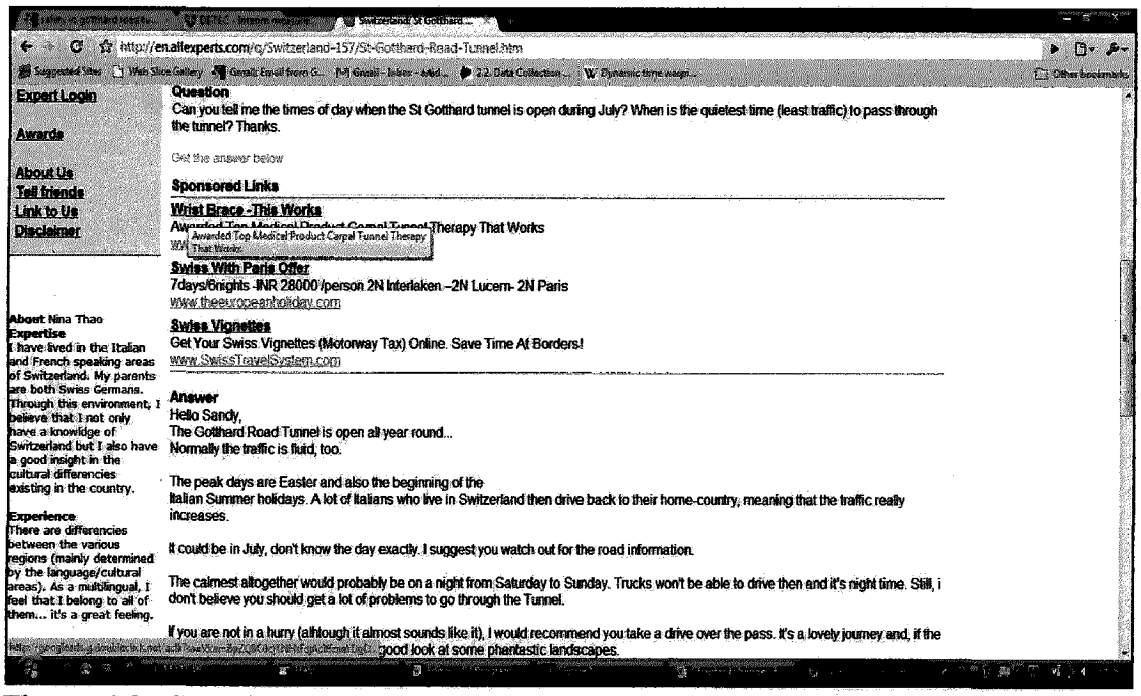


Figure 4.3: Snapshot of the site www.en.all.experts.com which gives information about the traffic condition in Gotthard tunnel all round the year.

Now we present the analysis of the dataset that is used in the proposed algorithm for detecting outliers.

Analysis to visualize outliers: The following graphs in the Figures 4.4 and 4.5 visualize outliers in the Gotthard tunnel dataset- number of motorcycles in one direction (in year 2005). The Table 4.2 explains the analysis shown in Figure 4.4 and Figure 4.5.

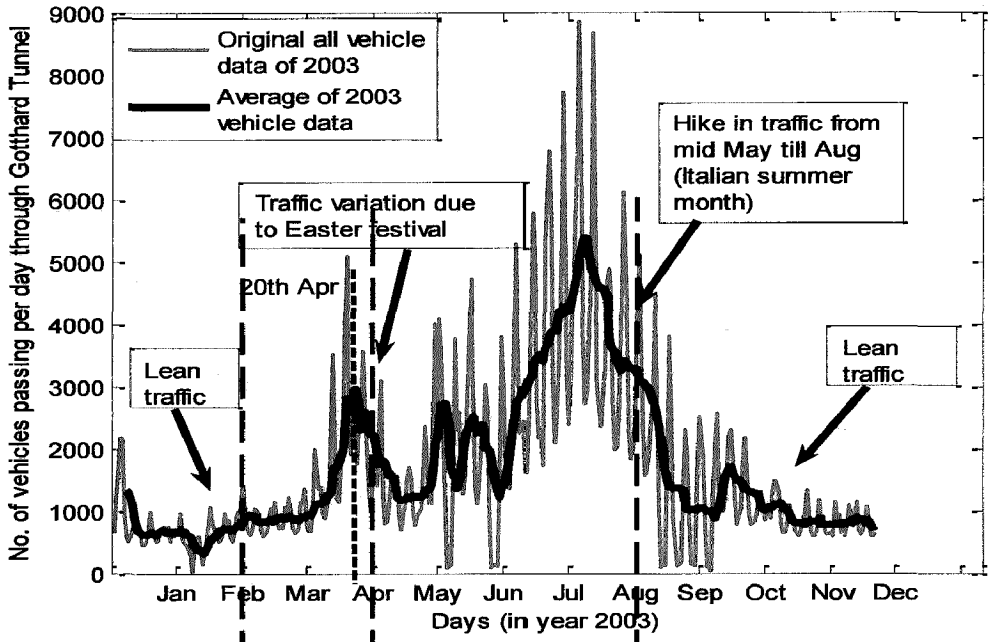


Figure 4.4: The Gotthard tunnel traffic of all types of vehicles in both direction in year 2

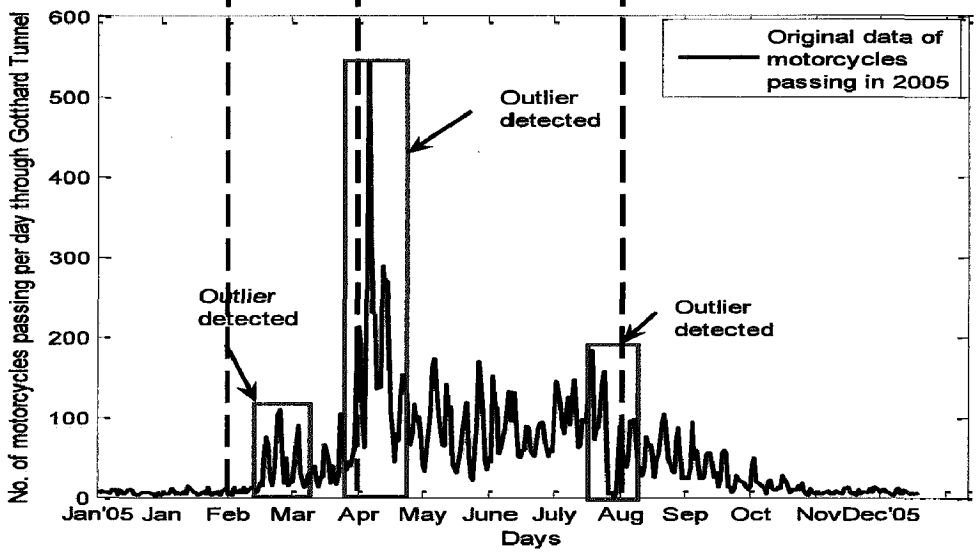


Figure 4.5: The original Gotthard traffic of motorcycles in one direction in year 2005

Table 4.2: The outliers detected in Gotthard tunnel dataset- number of motorcycles in one direction

Section of the graph explained	Traffic condition in year 2003	Traffic condition in year 2005
Section 1 (Jan to Feb)	Lean traffic	Lean traffic
Section 2 (Mar to Apr)	Traffic is high in last week of March due to Easter falling on 20th of April'03	**Outlier detected:** Traffic is high in last week of March due to Easter falling on 27th of March'05 **Outlier detected:** A very high rise is seen in the first few weeks of April.
Section 3 (May to Aug)	Hike in traffic	Hike in traffic **Outlier detected:** A sharp fall is seen in the last few weeks of August.
Section 4(Sept to Dec)	Lean traffic	Lean traffic

The next section presents the outliers detected by the proposed algorithm. The detailed analysis of the detected outliers is presented and the evidences are also given to prove the analysis done.

4.1.1 Results with Gotthard Tunnel Dataset: Number of Motorcycles in One Direction (in Year 2005).

The data buffer size is taken 30. The outlier subsequence length is taken 8. Since the data is being generated per day so we are detecting the anomalous weeks per month. The reference set size is also taken as 30.

In the following discussion we analyse each outlier detected by the proposed algorithm:

Outlier detected: Easter Festival 27th of March 2005

Easter and the holidays that are related to it are moveable feasts, in that they do not fall on a fixed date in the Gregorian or Julian calendars (which follow the motion of the sun and the seasons). Instead, they are based on a lunar calendar. The Easter Rule, which stated that Easter shall be celebrated on the first Sunday that occurs after the first full moon on or after the vernal equinox. The ecclesiastical "vernal equinox" is always on March 21. Therefore, Easter can be celebrated as early as on March 22 or as late as on April 25 [20]. The Figure 4.6 shows the outlier detected due the Easter festival on 27th March 2005. The following Table 4.3 presents the reason behind the excessive hike in traffic on 19th and 27th of March. The evidence of the reason for the detected outlier is presented in the Figure 4.7.

Analysis for the outlier detected

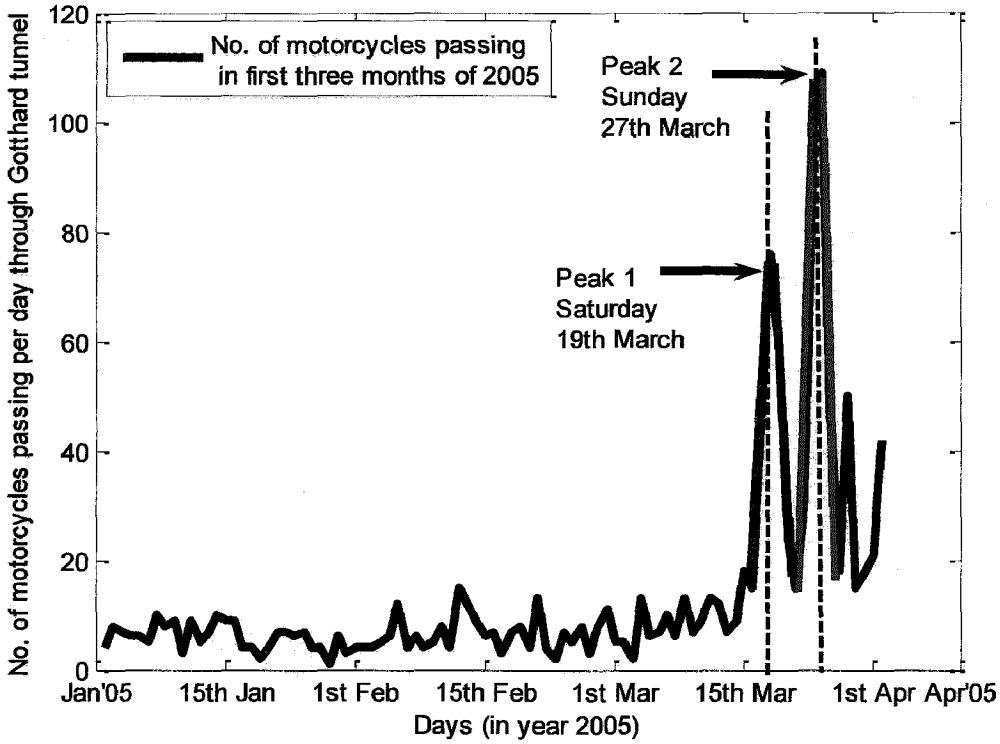


Figure 4.6: Outliers detected in the month of March for motorcycles passing in one direction in year 2005.

Table 4.3 presents the analysis for the detected outlier on 19th and 27th of March '05.

S.No.	Name	Analysis
1.	Peak 1	People who planned extented vacation, started travel a week befor the Easter festival.
2.	Peak2	People who moved in the same week of easter festival are high in number , that is planned a short vacation.

Evidence for the outlier detected: Easter festival was on 27th March in the year 2005.

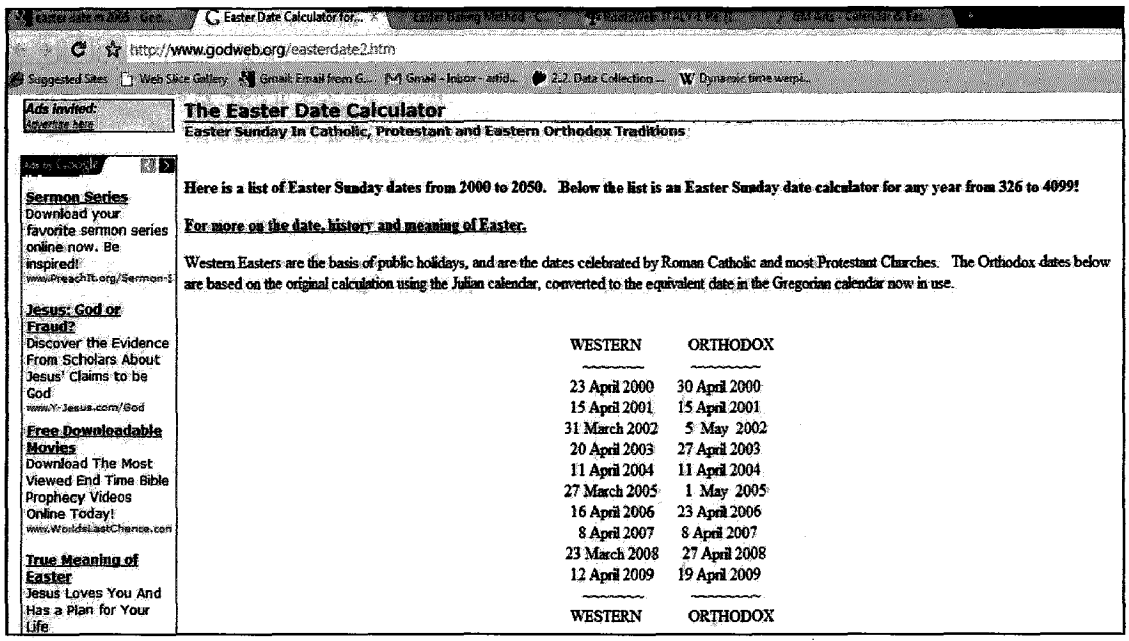


Figure 4.7: Snapshot of the site www.godweb.org gives information about the Easter date in year 2005.

Conclusion: So we can conclude that the outliers are detected correctly by the proposed algorithm. The figure 4.6 and Table 4.3 together explains the sudden rise of traffic of motorcycles in one direction in Gotthard tunnel. Also Figure 4.7 presents the evidence that Easter was on 27th of March in year 2005, which makes our analysis more concrete.

Outlier detected: May 3-14, 2005 UEFA European Under-17 Football Championship in Italy

The 2005 UEFA European Under-17 Football Championship was the fourth edition of UEFA's European Under-17 Football Championship. Italy hosted the championship, during May 3-14, 2005 [23]. The Union of European Football Associations (UEFA) represents the national football associations of Europe, runs Europe wide national and club competitions, and controls the prize money, regulations and media rights to those competitions. UEFA is the biggest of six continental confederations of FIFA [23]. The following Figure 4.8 and Table 4.4 present the analysis of the detected outliers. Also the evidence of the given analysis is shown in Figure 4.9, Figure 4.10 and Figure 4.11.

Analysis for the outlier detected

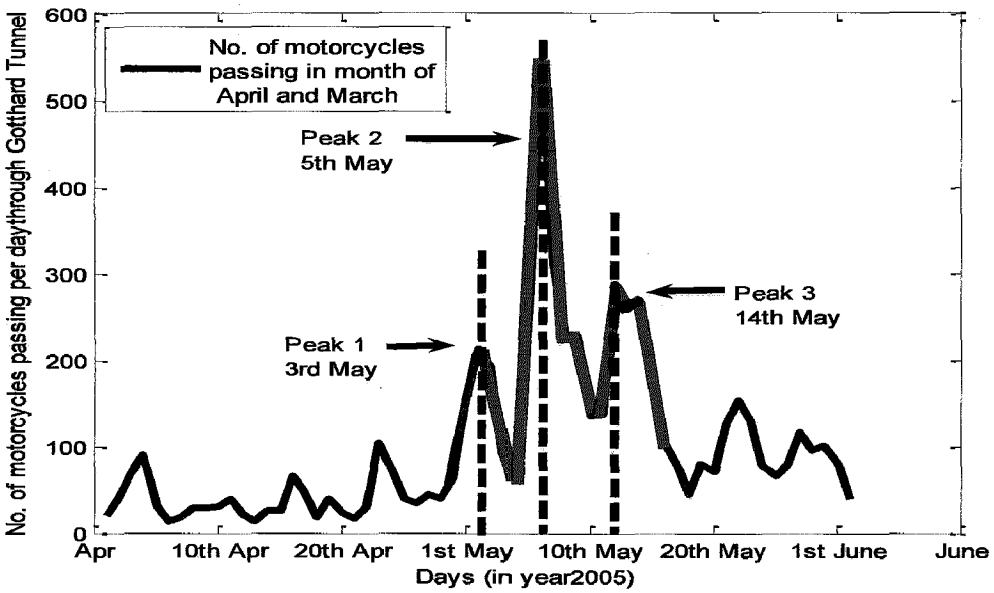


Figure 4.8: Outliers detected in month of May for motorcycle passing in one direction through Gotthard tunnel.

The following Table 4.4 presents the reasons behind the development of three peaks on dates 3rd, 5th and 14th of May 2005, shown in Figure 4.8.

Table 4.4: Presents the reason for the three peaks on 3rd, 5th and 14th of May 2005.

S.No.	Name	Analysis
1.	Peak 1	The UEFA under-17 football championship started on 3rd May. Group matches were scheduled, between: Belarus & England, Italy & Turkey, Isreal & Switzerland, croatia & Netherlands
2.	Peak2	On 5th May, the group matches were between: Italy & Belarus, Turkey & England, Switzerland & Netherland, Isreal & Croatia.
3.	Peak3	Final was on 14th May, between Netherland & Turkey.

Now we give the evidences that the UEFA under-17 championship started on 3rd May'05 in Italy. Then we present the evidence for the group matches held on 3rd and 5th May'05. After that we present evidence for the final match on 14th May.

Evidence: UEFA under-17 football championship started on 3rd May'05 in Italy

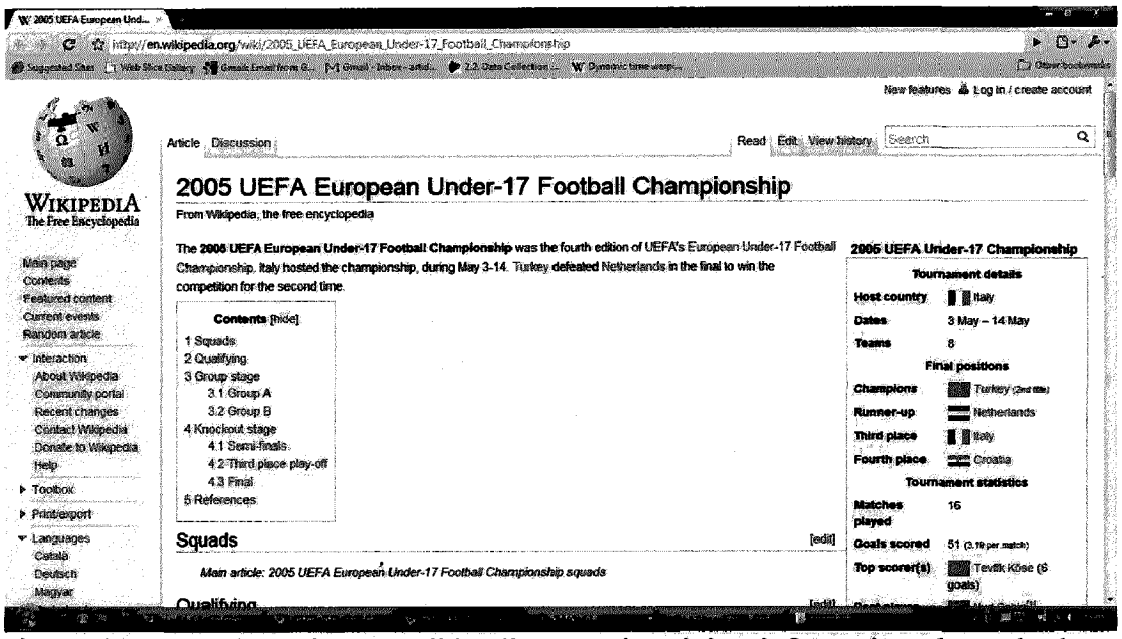
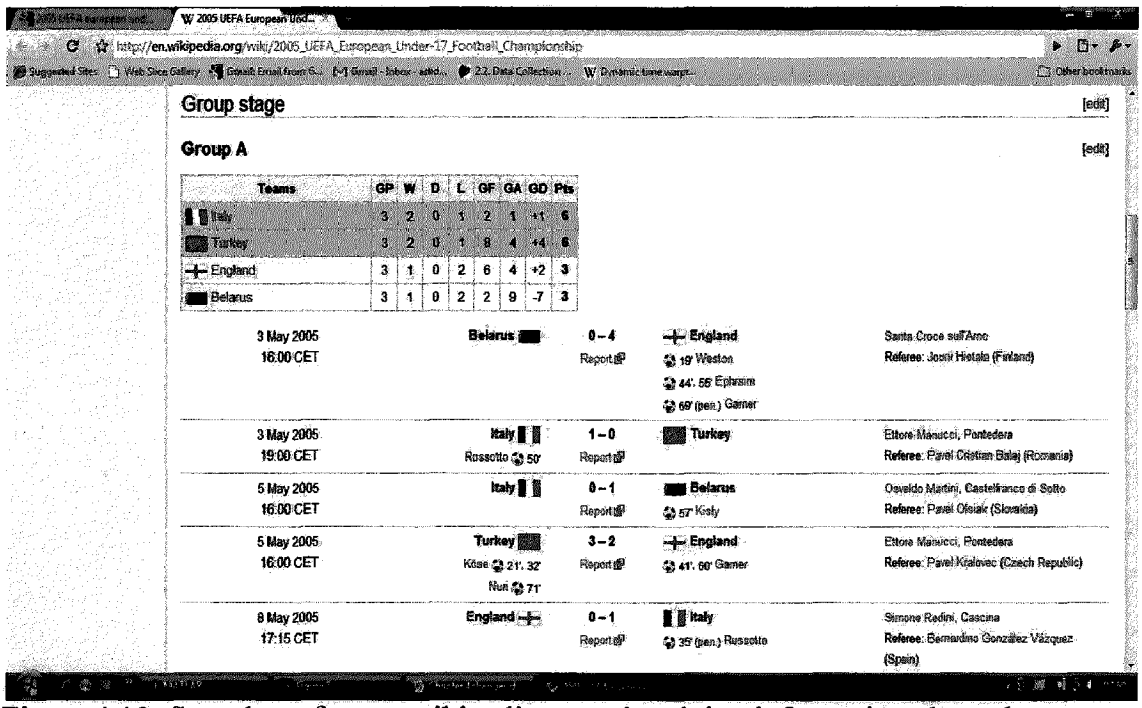


Figure 4.9: Snapshot of www.wikipedia.com site giving information about the host and date of UEFA under-17 football championship.

Evidence: Group matches that held on 3rd and 5th May'05



Group stage [edit]

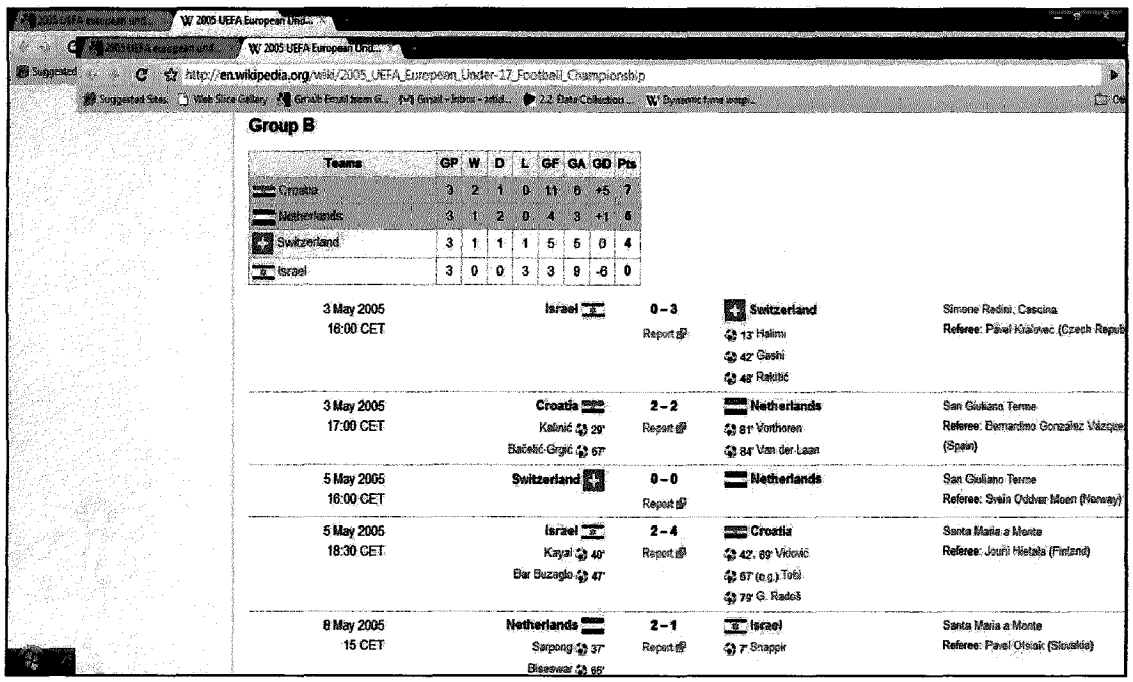
Group A [edit]

Teams	GP	W	D	L	GF	GA	GD	Pts
Italy	3	2	0	1	2	1	+1	6
Turkey	3	2	0	1	8	4	+4	6
England	3	1	0	2	6	4	+2	3
Belarus	3	1	0	2	2	9	-7	3

3 May 2005 16:00 CET	Belarus	0 – 4 Report	England 19' Weston 44', 56' Ephraim 69' (pen.) Garner	Santa Croce sull'Arno Referee: Jouni Hietala (Finland)
3 May 2005 19:00 CET	Italy Rossetto 50'	1 – 0 Report	Turkey	Ettore Mannucci, Pontedera Referee: Pavel Cristian Balaj (Romania)
5 May 2005 16:00 CET	Italy	0 – 1 Report	Belarus 57' Kisly	Osvaldo Martini, Castelfranco di Sotto Referee: Pavel Otsiak (Slovakia)
5 May 2005 16:00 CET	Turkey Kose 21', 32' Nas 71'	3 – 2 Report	England 41', 60' Garner	Ettore Mannucci, Pontedera Referee: Pavel Kralovec (Czech Republic)
8 May 2005 17:15 CET	England	0 – 1 Report	Italy 35' (pen.) Rossetto	Simone Redini, Cascina Referee: Bernardino Gonzalez Vazquez (Spain)

Figure 4.10: Snapshot of www.wikipedia.com site giving information about the group matches on 3rd and 5th May'05.

Evidence: Group matches that held on 3rd and 5th May'05



Group B

Teams	GP	W	D	L	GF	GA	GD	Pts
Croatia	3	2	1	0	11	6	+5	7
Netherlands	3	1	2	0	4	3	+1	5
Switzerland	3	1	1	1	5	5	0	4
Israel	3	0	0	3	3	9	-6	0

3 May 2005 16:00 CET	Israel	0 – 3 Report	Switzerland 13' Halimi 42' Gashi 48' Ralidic	Simone Redini, Cascina Referee: Pavel Kralovec (Czech Repub
3 May 2005 17:00 CET	Croatia Kalinić 29' Bačelić-Grgić 67'	2 – 2 Report	Netherlands 81' Vorthoren 84' Van der Laan	San Giuliano Terme Referee: Bernardino Gonzalez Vazquez (Spain)
5 May 2005 16:00 CET	Switzerland	0 – 0 Report	Netherlands	San Giuliano Terme Referee: Svein Oddvar Moen (Norway)
5 May 2005 18:30 CET	Israel Kayal 40' Bar Buzaglo 47'	2 – 4 Report	Croatia 42', 69' Vidović 67' (o.g.) Tobi 79' G. Rados	Santa Maria a Monte Referee: Jouni Hietala (Finland)
8 May 2005 15 CET	Netherlands Sarpong 37' Bisseswar 65'	2 – 1 Report	Israel 7' Snapir	Santa Maria a Monte Referee: Pavel Otsiak (Slovakia)

Figure 4.11: Snapshot of www.wikipedia.com site giving information about the group matches on 3rd and 5th May'05

Evidence: Finals on 14th of May'05

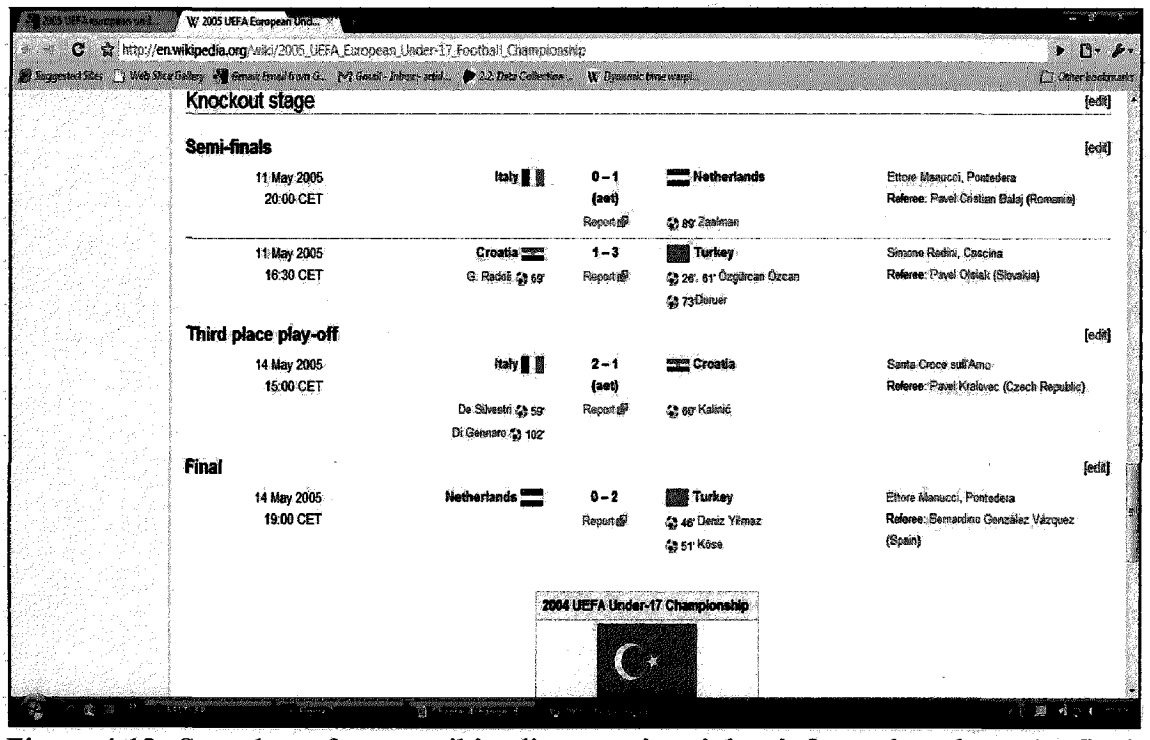


Figure 4.12: Snapshot of www.wikipedia.com site giving information about the final match on 14 May'05.

Conclusion: So we can conclude that the outlier is detected correctly by the proposed algorithm. The figure 4.8 and Table 4.4 together explains the sudden rise of traffic of motorcycles in one direction in Gotthard tunnel. Also Figure 4.9, Figure 4.10, Figure 4.11 and Figure 4.12 presents the evidence that UEFA under-17 championship was on 3rd May in year 2005 in Italy, which makes our analysis more concrete.

Outlier detected: June 15-29, 2005 FIFA Confederations Cup in Germany

Confederations Cup football tournament was the seventh FIFA Confederations Cup. It was held in Germany between 15 June and 29 June 2005 [23]. The following Figure 4.13 and the Table 4.5 present the analysis of the detected outlier. The Figure 4.14 and Figure 4.15 are used to present the evidence for the analysis done.

Analysis for the outlier detected

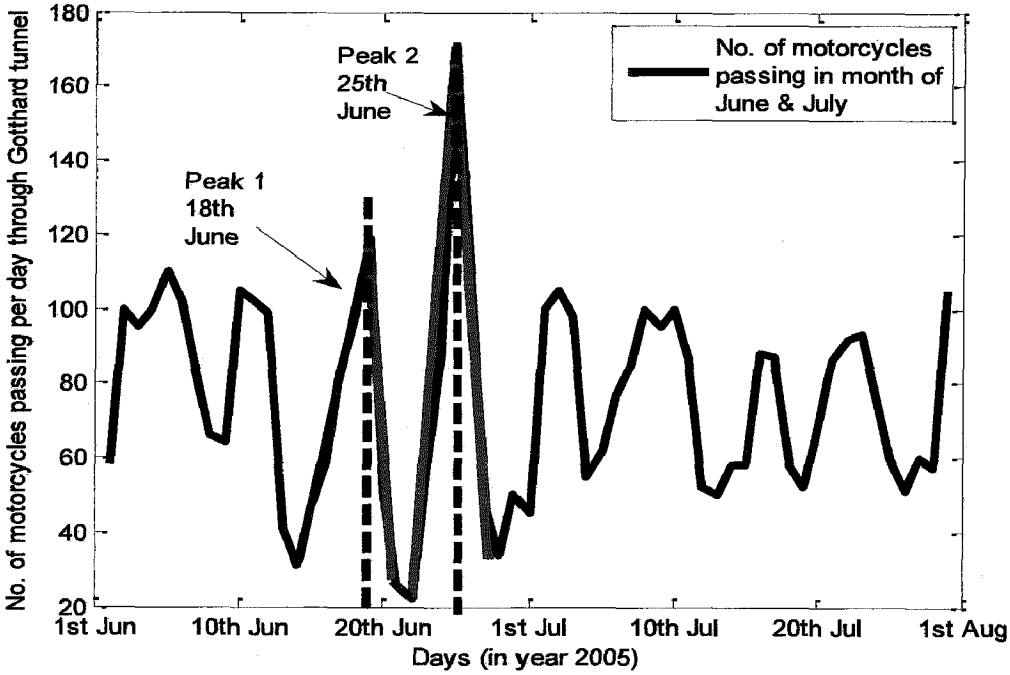


Figure 4.13: Outliers detected in month of June for motorcycle passing in one direction through Gotthard tunnel.

The following Table 4.5 presents the reasons behind the development of peaks peaks on dates 18th and 25th of June 2005, shown in Figure 4.13.

Table 4.5: Presents the reason for the two peaks on 18th and 25th of June 2005.

S.No.	Name	Analysis
1.	Peak 1	The group match of FIFA football championship on 18th of June were scheduled, between: Tunisia & Germany, Australia & Argentina
2.	Peak2	On 25th June, the semifinal was held between: Germany and Brazil .

Now we give the evidences that the FIFA confederation cup started on 15th June'05 in Germany. Then we present the evidence for the group match held on 18th of June and semifinal on 25th June.

Evidence: FIFA confederation cup started on 15th June to 29th June in Germany

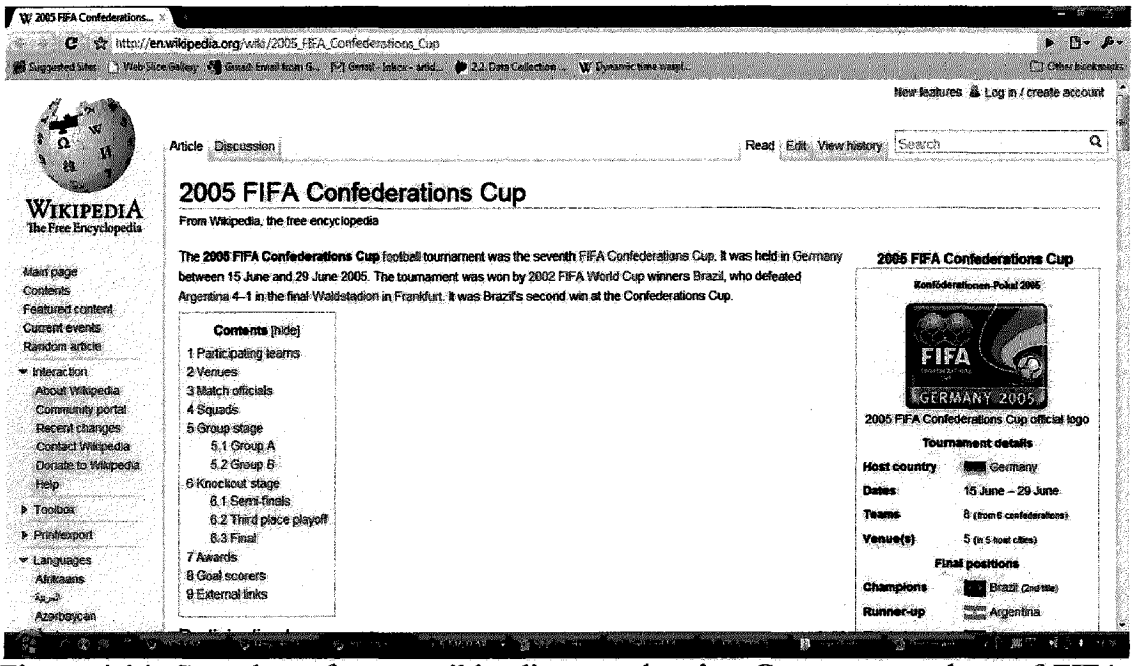


Figure 4.14: Snapshot of www.wikipedia.com showing Germany was host of FIFA Confederation cup 2005.

Evidence: Group match that held on 18th June'05

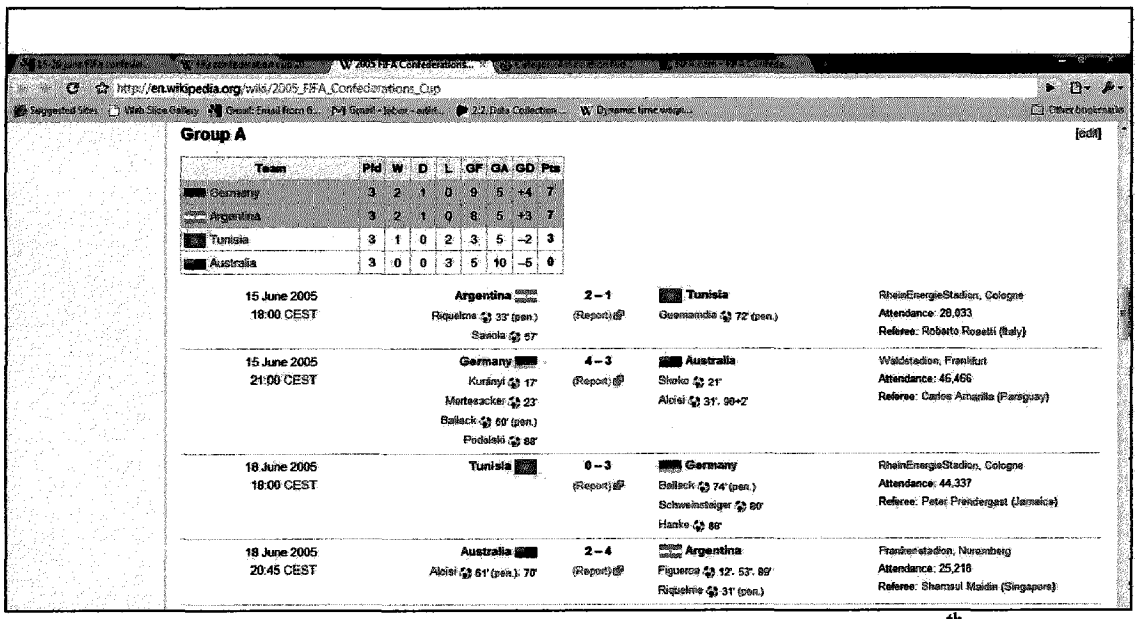


Figure 4.15: Snapshot of wikipedia site showing group match held on 18th of June'05.

Evidence: Semi-final match that held on 25th June'05

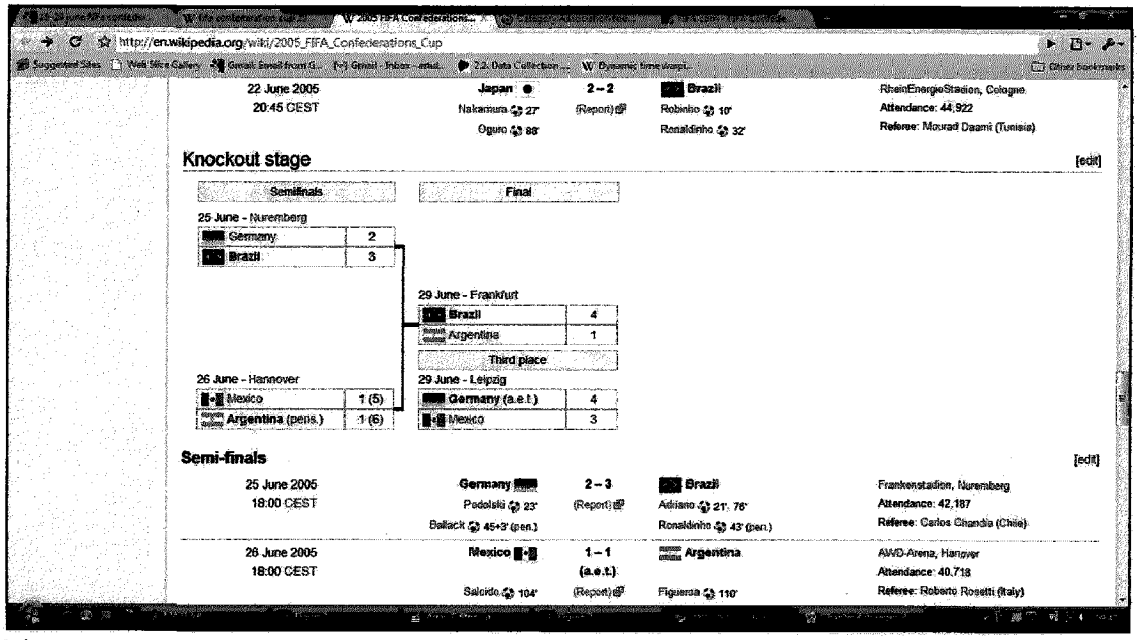


Figure 4.16: Snapshot of www.wikipedia.com showing the semifinal match held on 25th of June'05.

Conclusion: So we can conclude that the outlier is detected correctly by the proposed algorithm. The figure 4.13 and Table 4.5 together explains the sudden rise of traffic of motorcycles in one direction in Gotthard tunnel. Also Figure 4.14, Figure 4.15 and Figure 4.16 presents the evidence that FIFA confederation cup 2005 was on 15th June year 2005 in Germany, which makes our analysis more concrete.

Outlier detected: Heavy rainfalls in Switzerland from Saturday August, 20th to Monday August, 22nd, 2005.

The most heavy rainfalls since more than 100 years in Switzerland in 2005 (and ever since precise statistics are available) from Saturday August, 20th to Monday August, 22nd have devastated large regions of the country [24]. The prealpine regions Entlebuch (west of Lucerne), Obwalden and Bernese Oberland and the roads and railroads between Zurich / Lucerne and St. Gotthard at the foot of prealpine Mount Rigi were hit by landslides and rivers Schächen and Reuss flooded the region. The following Figure 4.17 and Table 4.6 present the analysis of the detected outlier. Also the evidence of the given analysis is shown in Figure 4.18.

Analysis for the outlier detected

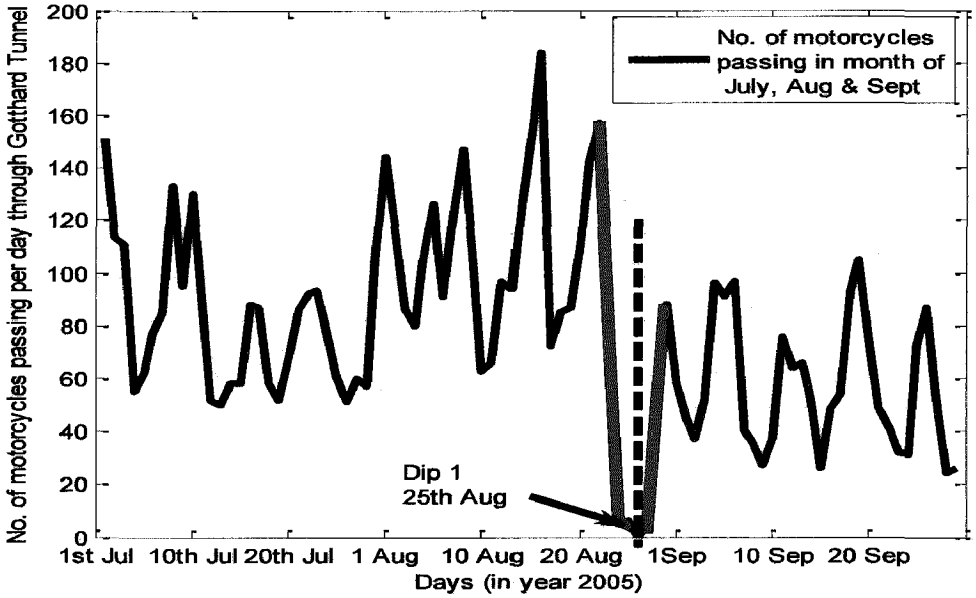


Figure 4.17: Outlier detected in month of August for motorcycle passing in one direction through Gotthard tunnel.

Table 4.6: Presents the reason for the dip on 25th of Aug'05

S.No.	Name	Analysis
1.	Dip 1	Due to heavy rainfall and landsides on 20 to 22 Aug all traffic routes were severly damaged and so 25th of Aug saw a severe dip in traffic.

Now we present the evidence that heavy rainfall and landslides in Switzerland on 20th Aug to 22nd Aug 2005, hit the transportation system severely.

Evidence: Heavy rainfall and landslides in Switzerland on 20th Aug to 22nd Aug'05

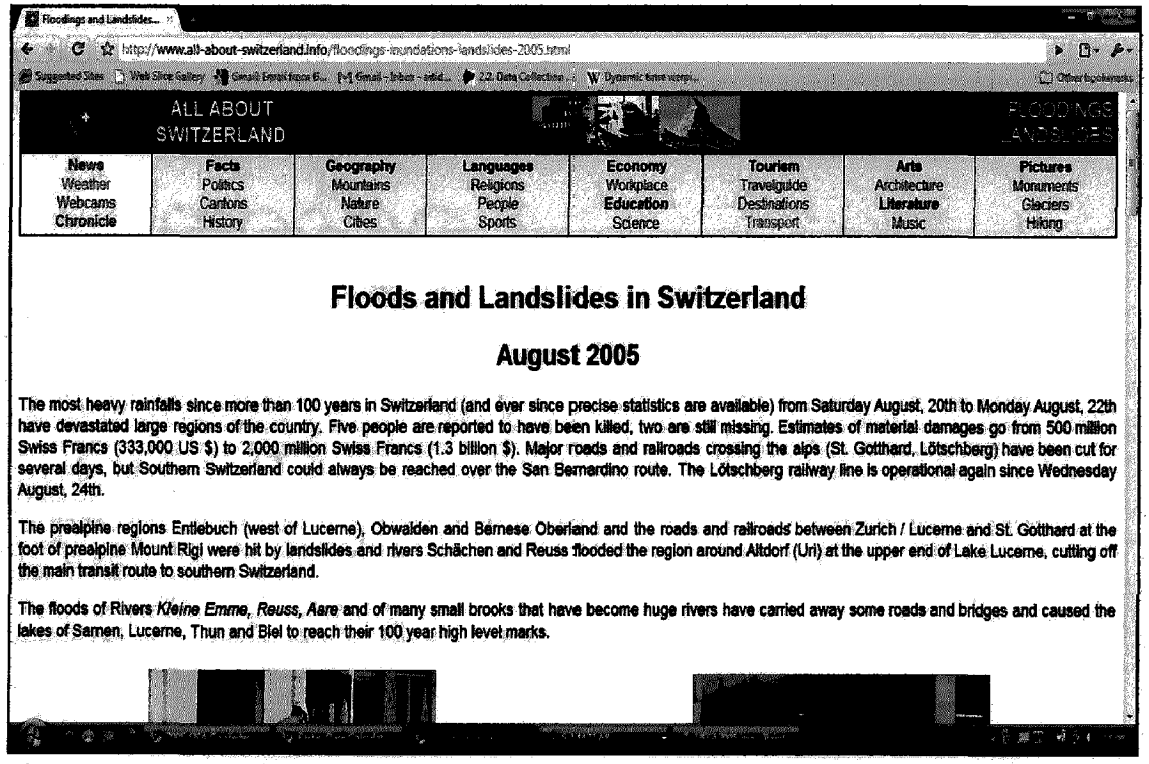


Figure 4.18: Snapshot of site www.all-about-swtizerland.info provides information about the devastating floods and landslides in Swtizerland in August 2005.

Conclusion: So we can conclude that the outlier is detected correctly by the proposed algorithm. The figure 4.17 and Table 4.6 together explains the sudden dip of traffic of motorcycles in one direction in Gotthard tunnel. Also Figure 4.18, presents the evidence that heavy rainfall and landslides really occurred, which makes our analysis more concrete.

4.2 ECG Dataset- Itstdb_20221_43

The dataset is taken from UCR Time Series Data Mining Archive. The dataset has been used for discord discovery and pattern matching algorithms [19]. It contains 3000 records.

Analysis of the ECG dataset

The electrocardiogram (ECG) is a diagnostic tool that measures and records the electrical activity of the heart in exquisite detail. Interpretation of these details allows diagnosis of a wide range of heart conditions. These conditions can vary from minor to life threatening. The ECG records this electrical activity and depicts it as a series of graph-like tracings, or waves. The shapes and frequencies of these tracings reveal abnormalities in the heart's anatomy or function [19]. The Figure 4.19 shows the graph of the given ECG dataset.

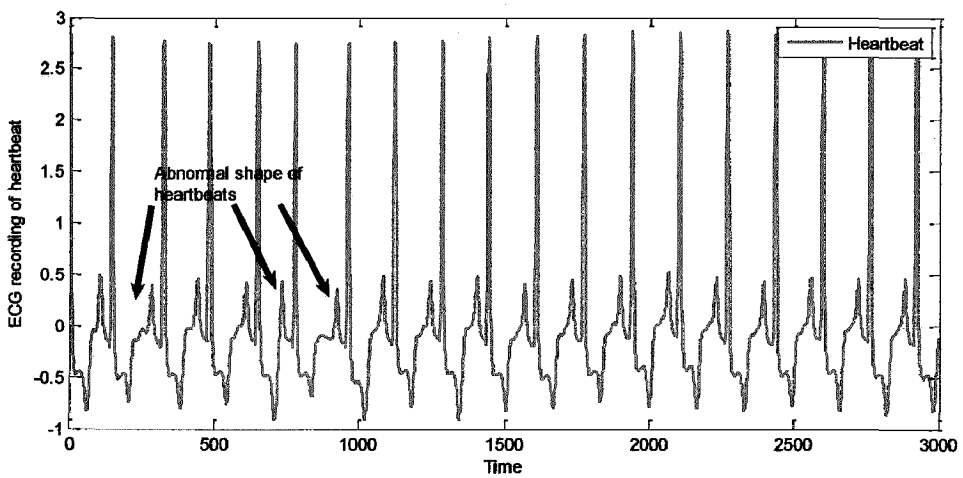


Figure 4.19: The graph shows the ECG reading of a heart with three abnormal beats.

4.2.1 Results with ECG Dataset-Itstdb_20221_43

Detected outliers

The ECG dataset-Itstdb_20221_43 time series is used here as streaming time series. The data buffer size is taken 500. The outlier subsequence is taken 150, since the length of one heart beat is approximately 150. The proposed algorithm detects the adaptive outliers as the anomalous heart beats with irregular shape and size. The Figure 4.20 shows that the three abnormal heartbeats were detected as outliers.

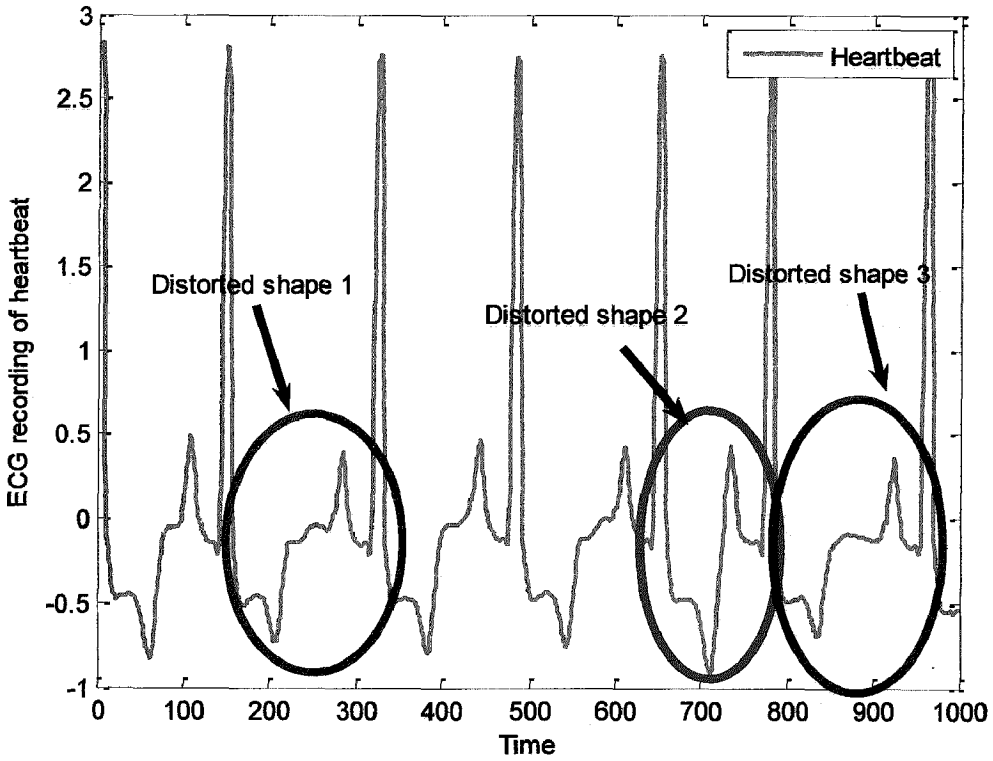


Figure 4.20: Three abnormal heartbeats were detected in received streaming ECG data

Chapter 5: Conclusion and Future Work

5.1 Conclusion

In this dissertation work, we considered the problem of detecting adaptive outliers in streaming time series data. After the completion of the dissertation work, we have reached the following conclusions about our proposed algorithm:

- The HOT SAX algorithm has been extended successfully to detect the local outlier subsequences in the streaming time series data.
- The outliers has been classified successfully into global and local outliers by adaptive rule-based classifier.
- The type has also been established successfully that is, "above normal" or "below normal" for each outlier classified as global.
- The degree of "outlierness" has also been identified successfully in terms of severe and mild abnormal behaviour.

5.2 Future Work

Some suggestions for future work in the proposed algorithm are as follows:

- The proposed algorithm can be extended to detect the outlier from other types of data such as, multidimensional or categorical data.
- The other distance measures can be used, instead of Euclidean distance measure, to evaluate the distance between the two subsequences in HOT SAX algorithm for detecting local outliers.
- Various other existing outlier detection techniques can be extended, just like HOT SAX algorithm, for detecting local outliers.

References

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, American Association for Artificial Intelligence, 1996.

[2] Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, second edidtion, Elsevier publication, 2008.

[3] Charu C. Aggarwal, A Framework for Diagnosing Changes in Evolving Data Streams, pages: 575-586, *in proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003.

[4] Feng Han, Yan-Ming Wang, Hua-Peng Wang, ODABK: An Effective Approach to Detecting Outlier in Data Stream, *in proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian*, 2006.

[5] Jessica Lin, Eamonn Keogh, Stefano Lonardi Bill Chiu., A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, *in proceeding of 8th workshop on Research issues in data mining and knowledge discovery ACM SIGMOD*, pages: 2-11, 2003.

[6] Barret, V., Lewis, T., Outliers in Statistical Data. *Wiley, Chichester*, 2001.

[7] Ng, R.T., Han, J., Efficient and Effective Clustering Methods for Spatial Data Mining, *In: VLDB*, pp. 144–155, 1994.

[8] Ester, M., Kriegel, H.P., Xu, X., A Database Interface for Clustering in Large Spatial Databases. *In: KDD*, pp. 94–99, 1995.

[9] Zhang, T., Ramakrishnan, R., Livny, M., BIRCH: An Efficient Data Clustering Method for Very Large Databases, *In: SIGMOD*, pp. 103–114, 1996.

[10] Sheikholeslami, G., Chatterjee, S., Zhang, A., WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases, *In: VLDB*, pp. 428–439, 1998.

[11] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, *In: SIGMOD*, pp. 94–105, 1998.

[12] Kozue Ishida,Hiroyuki Kitagawa, "Detecting Current Outliers: Continuous Outlier Detection over Time-Series Data Streams", volume 5181,pages 255-268, *Springer-Verlag Berlin Heidelberg*, 2008.

[13] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J., LOF: Identifying Density-Based Local Outliers., *In: SIGMOD*, pp. 93–104, 2000.

[14] Ke Zhang, Marcus Hutter, Huidong Jin, A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data, volume 5476, pages 813-822, *Springer Berlin*, 2009.

[15] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David Wai-Lok Cheung, Enhancing effectiveness of outlier for low density patterns, pages 535-548, *in PAKDD*, 2002.

[16] Hongqin Fan, Osmar R. Zaiane, Andrew Foss, Junfeng Wu, "A nonparametric outlier detection for effectively discovering top-n outlier from engineering data", pages 557-566, *in PAKDD*, 2006.

[17] Eamonn Keogh, Jessica Lin, Ada Fu, HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequences, In proceedings of the Fifth IEEE International Conference on Data Mining, pages 226-233, 2005.

[18] http://www.kdd.org/

[19] http://www.cs.ucr.edu/

[21] http:// en.all.experts.com/q/Switzerland-157/St-Gotthard-Road-Tunnel

[22] http://festivals.iloveindia.com/easter/when-easter

[23] http://wikipedia.com/

[24] http://www.all-about-swtizerland.info/floodings-landslides

List of Publication

[1] Shachi Yadav, Durga Toshniwal, "Dynamic Time warping on Symbolic Representation of Time Series", third IEEE International Conference on Computer Science and Information Technology, July 2010 (accepted).