Time Series Data Mining Tool

Samir Sheriff
Computer Science and Engineering
RV College of Engineering
Bangalore.
Email: samiriff@gmail.com

Satvik Neelakant
Computer Science and Engineering
RV College of Engineering
Bangalore.
Email: nsatvik@gmail.com

Vaishakh BN
Computer Science and Engineering
RV College of Engineering
Bangalore.
Email: vaishakhbn@gmail.com

***Abstract*—A time series is a sequence of data points, measured typically at successive points in time spaced at uniform time intervals. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data.In the context of statistics,the primary goal of time series analysis is forecasting. In the context of signal processing it is used for signal detection and estimation, while in the context of data mining, pattern recognition and machine learning time series analysis can be used for clustering, classification, query by content, anomaly detection as well as forecasting. This project is aimed making a time series data mining tool which can be used to accomplish the above goals.**

The tool developed can be used to perform anomaly detection, forecasting, similarity detection. The tool has been developed and tested using these data sets.

***Index Terms*—time series, forecasting, similarity detection, anomaly detection.**

I. Introduction

A time series is a set of observations X_t , each one being recorded at a specific time t. Discrete-time time series is one in which the set T of times at which observations are made is a discrete set. Continuous-time time series are obtained when observations are recorded continuously over some time interval, e.g., when T_0 belongs [0,1]. Examples of time series are the daily closing value of the ECG readings and the annual flow volume of the Nile River at Aswan. Time series are very frequently plotted via line charts. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

A. Literature Survey

A time series is a collection of observations made sequentially through time. At each time point one or more measurements may be monitored corresponding to one or more attributes under consideration. The resulting time series is called univariate or multivariate respectively. In many cases the term sequence is used in order to refer to a time series, although some authors refer to this term only when the corresponding values are non-numerical. Throughout this paper the terms sequence and time series are being used interchangeably. The most common tasks of time series data mining methods are: indexing, clustering, classification, novelty detection, motif discovery and rule discovery. In most of the cases, forecasting is based on the outcomes of the other tasks. A brief description of each task is given below.

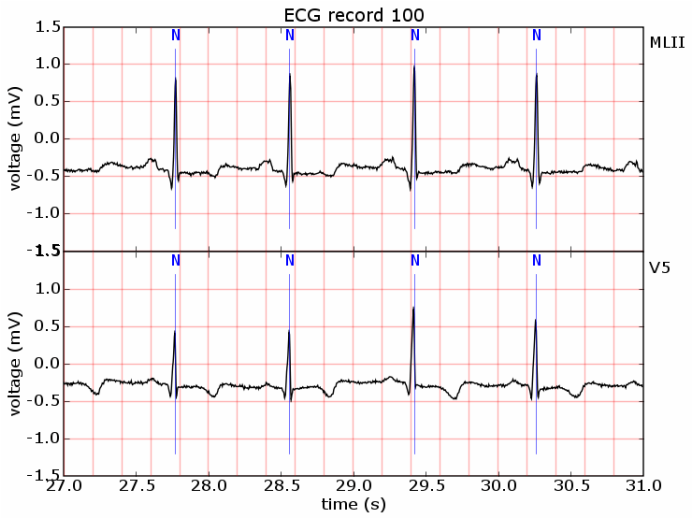

Fig. 1. Time Series Example: ECG

Indexing: Find the most similar time series in a database to a given query time series.
Clustering: Find groups of time series in a database such that, time series of the same group are similar to each other whereas time series from different groups are dissimilar to each other.
Classification: Assign a given time series to a predefined group in a way that is more similar to other time series of the same group than it is to time series from other groups.
Novelty detection: Find all sections of a time series that contain a different behavior than the expected with respect to some base model.
Motif discovery: Detect previously unknown repeated patterns in a time series database.
Rule discovery: Infer rules from one or more time series describing the most possible behavior that they might present at a specific time point (or interval).

The ability to model and perform decision modeling and analysis is an essential feature of many real-world applications ranging from emergency medical treatment in intensive care units to military command and control systems. Existing formalisms and methods of inference have not been effective in

real-time applications where trade-offs between decision quality and computational tractability are essential. The objective of this project is to fill the void that exists and help in proper analysis of time varying data.

B. SCOPE

The scope of a time series data mining tool is two fold. The first is to obtain an understanding of the underlying forces and structure that produced the observed data. The second is to fit a model and proceed to forecasting, monitoring or even feedback and feed forward control. The time series data mining tool can be used in the following fields.

- **Economic Forecasting**
- **Sales Forecasting**
- **Rainfall Analysis**
- **Stock Market Analysis**
- **Yield Projections**
- **Process and Quality Control**
- **Census Analysis**

C. METHODOLOGY

Time series analysis of data requires the user to able to view the different algorithms and the result obtained from each algorithm along with the graphs which help the user understand the time varying nature of the data. Hence, the representation of data becomes very important. Having understood this requirement in the early phase of the project, we adopted a methodology that will accomplish the objectives in a neat and intuitive way. A GUI was developed in the form of Java Server Pages and the back end was coded in Java which helped us exploit the object oriented paradigm in design of algorithms.

II. Software Requirements Specification

Software Requirement Specification (SRS) is an important part of software development process. It includes a set of use cases that describe all the interactions of the users with the software. Requirements analysis is critical to the success of a project.

A. Assumptions and Dependencies

- It is assumed that the user of this tool has basic understanding of time series data mining.
- Also, the user must have a decent knowledge of the interpretation of line graphs.

B. Specific Requirements

This section shows the functional requirements that are to be satisfied by the system. All the requirements exposed here are essential to run this tool successfully.

C. Functional Requirements

a) : The functionality requirements for a system describe the functionality or the services that the system is expected to provide. This depends on the type of software system being developed. The requirements that are needed for this project are :

- The data sets should be normalized so that the algorithms can be applied effectively.
- A good representation of the results should be made available to the users through proper representation media like graphs.

D. Software Requirements

1) DEVELOPERS MACHINE:

- Operating System: Windows 7/8, Linux, Mac
- Software Tools : Java, JDK 7.0, Apache Tomcat Server version 7.0
  Web Browser (Mozilla, IE8+, Chrome)
- IDE : Eclipse IDE for J2EE Developers
- API Libraries : JQuery UI and Ajax Libraries (Active Internet Connection)

2) END USERS MACHINE:

- Java Enabled Browser
- Active Internet Connection

E. HARDWARE REQUIREMENTS

- Processor: Intel Pentium 4 or higher version
- RAM: 512MB or more
- Hard disk: 5 GB

1) SOFTWARE INTERFACES : The Java Runtime Environment (JRE) is required to run the software.

III. SYSTEM ARCHITECTURE

This section provides an overview of the functionality and the working of the time series data mining tool. The overall functionality of the application is divided into different modules in an efficient way. The system architecture is shown in Figure 2

A. DATA FLOW DIAGRAMS

A DFD is a figure which shows the flow of data between the different processes and how the data is modified in each of the process. It is very important tool in software engineering that is used for studying the high level design.

There are many levels of DFDs. Level 0 gives the general description and level 1 gives the detailed description. Going higher in the level numbers greater description of the processes will be given.

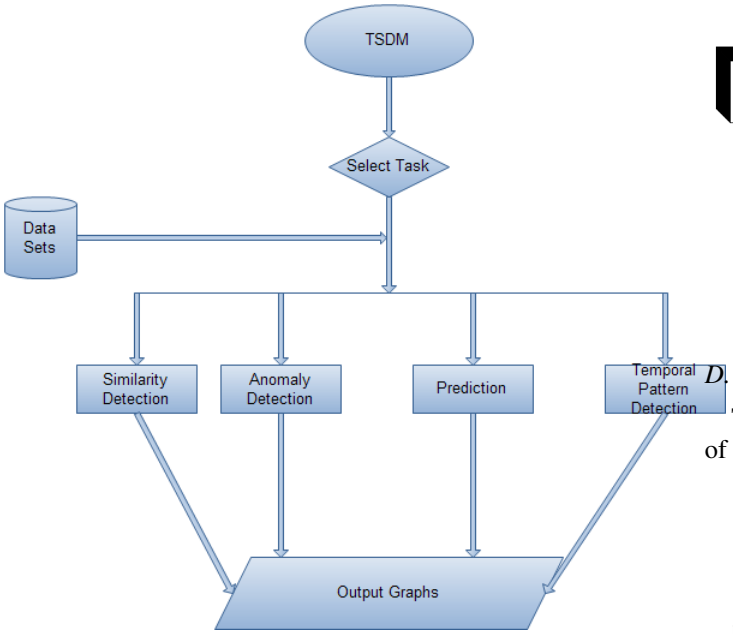

Fig. 2. System Architecture

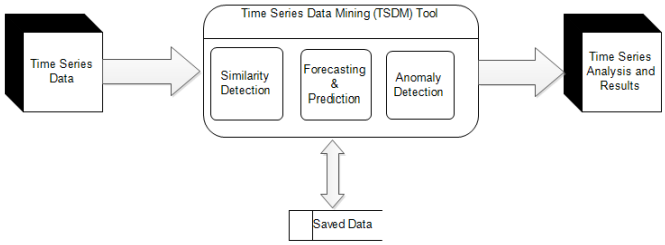

Fig. 3. DFD Level 0

B. DFD LEVEL 0

The level 0 DFD is shown in Fig. 3 below which gives the general operation of the TSDM Tool. There are two major components. One external entities called user and the TSDM Tool.

- User : The User is the one responsible to send the data/instructions to the TSDM Tool. The data may be a time series data or instructions to run datamining algorithms on the data.
- TSDM Tool : This tool contains various algorithms implemented under different categories like similarity detection, forecasting, anomaly detection etc. Depending on the instructions sent by the user, the algorithms is run and the results are sent back.

C. DFD Level 1

The figure 4 shows the DFD level 1 diagram.

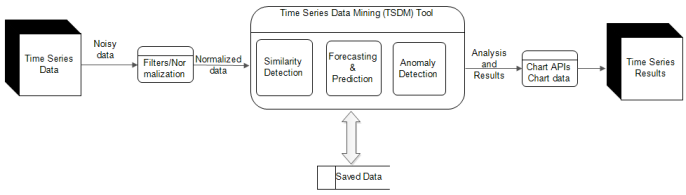

Fig. 4. DFD Level 1

D. User Interaction Diagram

The user interaction diagram in Figure 5 shows an overview of a user interacting with the TSDM Tool.

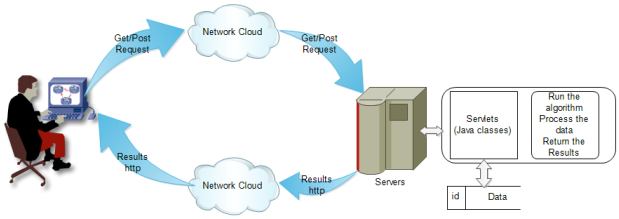

Fig. 5. User Interaction Diagram

IV. Implementation

The implementation phase of any project development is the most important phase and yields the final solution which solves the problem at hand.The implementation phase involves the actual materialization of the ideas, which are expressed in a suitable programming language. The factors concerning the programming language selection and platform chosen are described in the following sections.

A. Programming Language Selection

a) : The programming language chosen must reflect the necessities of the project to be completely expressed in terms of the analysis and the design documents. Therefore before choosing the language, features to be included in the project are decided. The time series data mining project needs the following features in a language to be implemented. Some of the features required are stated as follows:

- J2EE provides us with servlets and JSP which help in dynamically constructing web pages.
- J2EE provides us with Java Beans which help in proper data manipulation.
- JSP and servlets make use of Java backend in a very optimal manner. They have special tags which help us exploit these features.
- Java's core classes are designed from scratch to meet the requirements of an object oriented system.

With these necessities in mind, J2EE is selected as the optimal programming language to implement the project.

B. Platform

b) : The TSDM tool was built and designed on Windows Operating system family. They were specifically tested on Windows 7 with Google Chrome and Mozilla Firefox browsers. Because the product is browser based, any user with the browsers mentioned above will be able to run the tool. The product is hence platform independent in the true sense.

C. Modules

This project is not yet complete and is currently under development. The description of the modules are below :

- **Similarity Detection** : This module helps in finding similarity patterns (that occur at regular intervals in case of periodic time series), comparing different time series data. SAX and DTW are the main algorithms implemented/used in this module.
- **Forecasting and Prediction** : This module contain algorithms/models which can be trained from the past time series data and can be used to predict the future values of a time series.
- **Anomaly Detection** : This module contains algorithms that help in indicating anomalous patterns in the time series data analyzed. Anomalies are patterns in time series which deviate from the normal behavior and can indicate fraud/danger depending on the application. For example in an industry which produces the blades, the thickness of the blade can be monitored by a machine as a time series and any deviation from the normal error rate can signal an error in the manufacturing process.
- **Temporal Pattern Finder** : This module helps in finding hidden temporal patterns in a time series. This module can be further extended to implement clustering techniques.

V. Experimental Analysis and Results

As explained earlier, major module present in the project are as follows :

- Similarity Detection Module
- Prediction and Forecasting Module
- Anomaly Detection Module
- Temporal Pattern Finder Module

Since these modules are independent of each other, they have different evaluation metrics. The evaluation metrics used for performance analysis of each module is explained in the following sub sections.

A. Metrics for Similarity Detection Module

The algorithms implemented under this module are :

- **Dynamic Time Wrapping (DTW) Algorithm**
- **SAX Algorithms**

The evaluation metrics are :

- **Euclidean Distance** In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric.
  In Cartesian coordinates, if p = $(p_1, p_2, ..., p_n)$ and q = $(q_1, q_2, ..., q_n)$ are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by:

$$d(p,q) = d(q,p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=0}^{n} (p_i - q_i)^2}$$

  In the project, p and q can be visualized as two time series to be compared, where p_iś and q_iś are the time series values. Depending on the distance, the similarity of two or more time series with a give base series is found out. This metric is used in the DTW approach.
- **String Comparison** In SAX Algorithm, the time series is converted into a string of character as explained. Given two or more time series, which are represented by strings, a string comparison algorithm is run and the similarity is found out. There are various string comparison algorithms. In this project KMP algorithm has been used.

B. Metrics for Prediction and Forecasting Module

The algorithms implemented under this module for modeling and forecasting time series are :

- NARX-Neural Network
- Moving Average Forecaster
- Moving Geometric Average Forecaster
- Moving Exponential Average Forecaster

Modeling a time series is an regression problem, the evaluation metrics are :

- **Root-Mean-Square Deviation** - The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent.
  The RMSD of predicted values y_p for times t of a regression's dependent variable y is computed for n different predictions as the square root of the mean of the squares of the deviations:

$$RMSD = \sqrt{\sum_{i=0}^{n} (y_p - y)^2 / n}$$

The accuracies of different algorithms are compared and presented in the next section.

C. Metrics for Anomaly Detection Module

The anomaly detection algorithms require the controller/user to specify various parameters which determine the anomalous points in the time series.

Algorithms implemented under this module are :

- **Cumulative Sum Approach (CUSUM)**
- **Statistical Approach**

These algorithms require a **threshold value** to be specified by the user and depending on this value, anomalous data points are determined.

D. Temporal Pattern Finder Module

c) : In this module, a Genetic Algorithm has been implemented to optimize the algorithm. The fitness function used in the GA determine the accuracies of the patterns found. But eventually user intervention is required to interpret the resulting patterns detected by the algorithm. The results are documented in the next section.

VI. Experimental Dataset

The data sets considered in this project are

- Sea Level Dataset : Indicating the sea level at various times of a day.
- Water Level : Ground Water level data, indicating the ground water level during various moths of an year for upto 5 years.
- Finance Dataset : Consisting of stock index values of Nifty and Vix collected every minuted for a week.(5 days,during market hours).
- ECG Dataset : The ECG voltage values of patients collected every 4ms.(for 10 patients).

All the experiment analysis and results are presented using the **Water Level** data set. As explained earlier, some algorithms require certain parameters which determine their accuracies.

VII. Conclusion

A. Summary

In this project, we were able to successfully implement the Time Series Data Mining Tool for analyzing the time series and test its performance. This tool mainly contains four modules, they are - **Similarity Detection, Forecasting and Prediction, Anomaly Detection and Temporal Pattern Finder**, which we were successful in implementing and testing. The results obtained were presented in the previous section.

Initially this project mainly focused on analyzing the sea and water level time series. Later this application was extended to any uni-variate time series data. Users can upload the time series data to be analyzed and get the results instantly. Major data sets used were :

- Sea Level Dataset : Indicating the sea level at various times of a day.
- Water Level : Ground Water level data, indicating the ground water level during various moths of an year for upto 5 years.
- Finance Dataset : Consisting of stock index values of Nifty and Vix collected every minuted for a week.(5 days,during market hours).
- ECG Dataset : The ECG voltage values of patients collected every 4ms.(for 10 patients).

In this project, we also analyzed the efficiencies of different algorithms for the same tasks and also compared the results for different data sets.Clearly more work needs to be done.

VIII. Future enhancements

Some of the future enhancements are :

1) The size of the time series data analyzed is in terms of Mega Bytes. For larger dataset(In terms of GBs) or big data, distributed computing technologies like Hadoop can be used.
2) The application can be extended to analyze multi variate time series data.
3) The application could be made more responsive by using Threads and Parallel/Cloud Computing
4) One more extension could be analyzing twitter post data with respect to time and predicting the trends. This requires NLP, but is an example of time series.
5) Efficient algorithms using Support Vector Models (SVMs) for forecasting, Hidden Markov Model for anomaly detection can be implemented.
6) This application uses static time series data, enhancements can be made to use real time data.(In finance applications)
7) This application can be converted into an mobile application (android, iPhone, iPad) where the users can analyze the time series data on the go and share the results on facebook

Acknowledgment

The authors wish to express their gratitude to **Dr. Shobha G. and Mrs. Shantha Rangaswamy**, who offered invaluable assistance, support and guidance throughout the development of this tool.

References

[1] Agrawal R., Lin K.-I., Sawhney H. S., Shim K.,*Fast Similarity Search in the Presence of Noise,Scaling, and Translation in Time-Series Databases*

[2] Genetic Algorithm for constructing DT - http://www.jprr.org/index.php/jprr/article/viewFile/44/25

[3] Decision Trees - http://web.cecs.pdx.edu/~mm/MachineLearningWinter2010/pdfslides/DecisionTrees.pdf

[4] Project brief for the DT using Horse data sets -

[5] Hamilton, James *Time Series Analysis published by Princeton University Press, 1994*

[6] Ann Ratanamahatana; Chotirat, Lin; Jessica, Gunopulos; Dimitrios, Keogh;Eamonn Riverside, Vlachos;Michail, Das; Gautam - *Mining Time Series Data* University of California,IBM T.J. Watson Research Center, University of Texas, Arlington.

[7] Brown, Robert Goodell (1963). *Smoothing Forecasting and Prediction of Discrete Time Series.* Englewood Cliffs, NJ: Prentice-Hall.

[8] Kurt Pohlen *The Review of Economics and Statistics* Vol. 11, No. 3 (Aug., 1929), pp. 149-151 Published by: The MIT Press