*International Journal For Technological Research In Engineering*
*Volume 4, Issue 9, May-2017* *ISSN (Online): 2347 - 4718*

# OPTIMIZING PERFORMANCE OF CLUSTERING TIME SERIES DATABASE USING FAST FEATURE SELECTION ALGORITHM

Sandhya Devi[1], C.P Singh[2]
S.R Group Of Institutions (College Of Science & Engineering) Jhansi.

***Abstract: Nowadays, size of real world time series data sets can take trillion observations and even more. Data mining task is it to extract new meaningful information form this massive amount of data. Many techniques that are well known for data mining in cross sections have been implemented and developed; but time series data mining methods are not as established and satisfactory yet. Large time series also give arise to many problems like very high dimensionality and up to today, researchers haven't agreed on best practices in this regard. This research paper gives hybrid method for classification of large time series and the proposed problem. The positive results obtained by the designed classification framework for various performance measures indicate that the proposed methodology is useful to simplify the process of distance selection in time series clustering tasks.***

***Keyword: Relief algorithm, Multi-label Classification Time series database.***

## I. INTRODUCTION

The task of Classifying time series for pattern discovery has an objective to find out a set of model patterns or profiles that represent as faithfully as possible the original data set, in a way that every independent vector of this original data can be considered as one of the models submitted to acceptable deviations or drifts, or an outlier at the most. The difference between time series and normal classification is that, in the time series case, the shape of input vectors entails features that are arranged in time. Hence, in univariate time series an input vector is usually the succession of values that a certain variable takes throughout a specific time scope. Classifying time series is of two types: (a) raw-data-based, where clustering is directly applied over time series vectors without any space-transformation previous to the clustering phase. Several works concerning each kind of time series clustering are referred to in detail in [2]. (b) feature-based or model-based, i.e., previously summarizing or transforming raw data by means of feature extraction or parametric models, e.g., dynamic regression, ARIMA, neural networks [1]; so the problem is moved to a space where clustering works more easily; Beyond the obvious loss of information due to feature-based or model-based techniques, they can also present additional drawbacks; for instance, the application-dependence of the feature selection, or problems associated to parametric modeling. On the other hand, characteristic drawbacks of raw-data-based approaches are: working with high-dimensional spaces (curse of dimensionality [3]), and being sensitive to noisy input data.

Similarity Measure: We consider similarity as the measure that establishes an absolute value of resemblance between Two Vectors, in principle isolated from the rest of the vectors and without assessing the location inside the solution space. Considering continuous features, the most common metric is the Euclidean distance:

$$d_E(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})(\vec{x}-\vec{y})'}$$

Note that Euclidean distance is invariant when dealing with changes in the order that timefields/features are presented; it means that it is in principle blind to capture vector or feature correlation.For time series data comparison, where trends and evolutions are intended to be evaluated, or when the shape formed by the ordered succession of features (i.e., the envelope) is relevant, similarity measures based on Pearson's correlation:

$$d_C(\vec{x},\vec{y}) = 1 - \frac{(\vec{x}-\bar{\vec{x}})(\vec{y}-\bar{\vec{y}})'}{\sqrt{(\vec{x}-\bar{\vec{x}})(\vec{x}-\bar{\vec{x}})'}\sqrt{(\vec{y}-\bar{\vec{y}})(\vec{y}-\bar{\vec{y}})'}}$$

It have also been widely utilized, although it is not free of distortions or problems [4].

## II. PROPERTIES AND CHALLENGES OF TIME SERIES DATA

Before we come up with time series data mining methods, we itemize which problems need to be tackled. As a general rule, large time series come along with super-high dimensionality, noise along characteristic patterns, outliers and dynamism. Moreover, the most crucial challenge in time series data mining is the comparison of two or more time series which are shifted or scaled through time or in amplitude. The problems that need to be tackled in time series data mining arise from typical properties of large time series. Firstly, as one observation of a time series is viewed as one dimension, the dimensionality of large time series is typically very high.

The visualization alone of time series which are larger than a several ten thousand observations can be challenging Lin et al. (2005). Working with super-high dimensional raw data can be very costly with respect to processing and storage costs. Therefore, a high level representation of the data or abstraction is required. Besides, the basic philosophy ofdata mining implies that avoiding a potential information loss by studying the raw data is not convenient and too slow. In the context of time series data mining, noise along characteristic patterns are additive white noise components [6]. Provided that we are interested in global characteristics, the time series

data mining techniques need to be robust against noisy components. If such massive amounts of data are collected, the sensitivity towards Measurement errors and outliers can be high. At the same time, long time series enable us to better differentiate between outliers and rare outcomes. Rare outcomes which would be categorized as outliers in small subsamples help us to better understand heterogeneity.

## III. METHOD

Before jumping into actual data mining, it is essential to preprocess the data at hand. Firstly, large time series data is often very bulky. Thus, directly dealing with such data in its raw format is expensive with respect to processing and storage costs. Secondly, we are dealing with time series which are no more comprehensible with the unaided eye in its raw format. Therefore we beforehand reduce dimensionality or segment the time series and then index them. In the light of lacking natural clarity of the raw data, visualization techniques and tools for large time series emerged and are presented here. Moreover, similarity measures are the backbone of all data mining applications and need to be discussed.

Non Data Adaptive Representation Techniques: Non data adaptive representation techniques have always the same transformation parameters regardless the features of the data at hand. So, the transformation parameters are fixed a priori. One subgroup of non-data adaptive representation techniques are operating in the frequency domain. Their logic based on the basic idea of spectral decomposition:

Any time series can be represented by a finite number of trigonometric functions. Generally speaking, operating in the frequency domain is valid as the Euclidean distances between two time series is the same in the time domain and in the frequency domain and hereby preserve distances.

Data Adaptive Representation Techniques: Data adaptive representation techniques are (more) sensitive to the nature of the data at hand. The transformation parameters are chosen depending on the available data and not a priori as for non-data adaptive techniques. However, almost all non-adaptive techniques can be turned into data adaptive approaches by adding data-sensitive proceeding schemes.

## IV. PROPOSED METHOD FOR FEATURE SELECTION:

Feature selection is the process of selecting a subset of relevant features for use in model construction. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide nomore information than the currently selected features, and irrelevant features provide no useful information in any context.

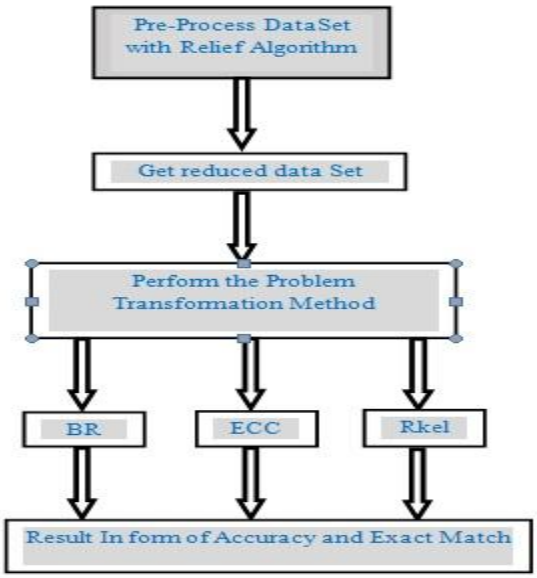



Fig.1: Proposed Model

Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points).The performance, robustness, and usefulness of classification algorithms are improved when relatively few features are involved in the classification. Thus, selecting relevant features for the construction of classifiers has received a great deal of attention. With the aim of choosing a subset of good features with respect to the target concepts, feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility.

Relief Algorithm:The Relief algorithm was first described by Kira and Rendell as a simple, fast, and effective approach to attribute weighing. The output of the Relief algorithm is a weight between −1 and 1 for each attribute, with more positive weights indicating more predictive attributes. The pseudo code for Relief is shown below. The weight of an attribute is updated iteratively as follows. A sample is selected from the data, and the nearest neighboring sample that belongs to the same class (nearest hit) and the nearest neighboring sample that belongs to the opposite class (nearest miss) are identified. A change in attribute value accompanied by a change in class leads up to weighting of the attribute based on the intuition that the attribute change could be responsible for the class change. On the other hand, a change in attribute value accompanied by no change in class leads to down weighting of the attribute based on the observation that the attribute change had no effect on the class. This procedure of updating the weight of the attribute is performed for a random set of samples in the data or for every sample in the data. The weight updates are then averaged so that the final weight is in the range [−1, 1]. The attribute weight estimated by Relief has a probabilistic interpretation. It is proportional to the difference between

two conditional probabilities, namely, the probability of the attribute's value being different conditioned on the given nearest miss and nearest hit respectively.

Chain Classifier Algorithm It involves Q-binary classifiers as in a BR method. It resolves the BR limitations, by taking into account the label correlation task. The classifiers are linked along a chain where each classifier deals with the BR problem associated with the label. Each link in the chain is expressed with the 0/1 label associations of all previous links. Proposed Algorithm steps are as follows:
TRAINING(D = {(x1, y1), . . . , (xN , yN )})
1 for j = 1…….L
2 do →the j th binary transformation and training
3 D'j ← {}
4 for (x, y) ∈ D
5 do x← [x1, . . . , xd , y1, . . . , yj−1]
D'j ←D'j∪ (x', yj)
7 →train hj to predict binary relevance of yj
8 hj: D'j → {0, 1}
Random K-label pruned set (RAkel) It constructs an ensemble of LP classifiers. It breaks the large label-sets into m models or subsets, which are associated with random and small sized k-label-sets. It takes label correlation into account and also avoids LP's problems within the large number of distinct label-sets. Given a new instance, it queries models and finds the average of their decisions per label. Also, it uses the threshold value t to obtain the final prediction. The final decision is positive for a specific label if the average decision is greater than the given threshold t. Thus, this method provides more accuracy of results

RAkel Algorithm
Input: Set of labels L of size M, training set D, label set size k
Output: Number of models m, k-label sets Ri, corresponding LP classifiers hi
m = [M/k] upper bound value
for i=1 to m
doRi= Null
for j=1 to k
do if L = Null then break;
Zi← randomly selected label from Li;
Ri←Ri U {Zi};
L ← L\{Zj}
Train an LP classifier hi based on D and Ri.

## V. IMPLEMENTATION AND EVALUATION

The experiments and results for the proposed classification and optimization based result are declared. Meka and Weka Tools are used for implementation. The Proposed Classification methodology has been developed for the training process. First the datasets used for the evaluation is described and then a large training data set for train the proposed model. Finally the experimental results are presented and discussions about the performance of the methods are given.

Explanation of Dataset:

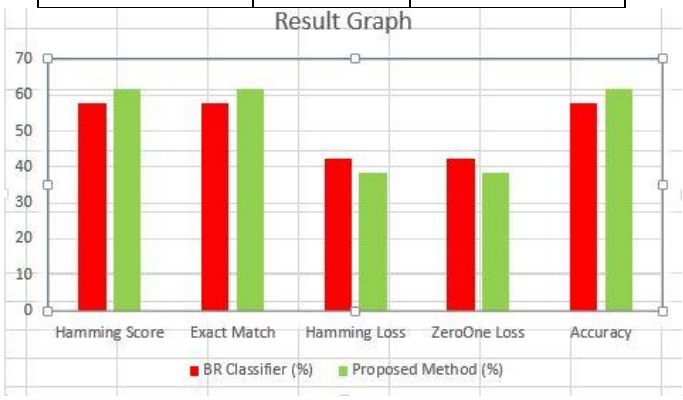
| Data Set | F | L |
|---|---|---|
| Experiment-1 | 4 | 3 |
| Experiment-2 | 103 | 14 |
| Experiment-3 | 71 | 6 |

Model characteristic:

| CV Folds | 10 |
|---|---|
| Threshold | PCut1 |
| Verbosity | 3 |
| Test File | Experiment test file |

Performance Measures Result Through Br And Proposed Methodology

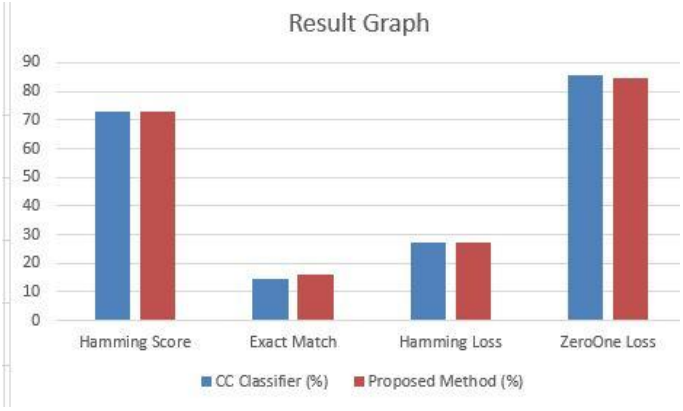
| Performance Measure | BR | Proposed Method |
|---|---|---|
| Hamming Score | 57.8 | 61.8 |
| Exact Match | 57.8 | 61.8 |
| Hamming Loss | 42.2 | 38.2 |
| ZeroOne Loss | 42.2 | 38.2 |
| Accuracy | 57.8 | 61.8 |




Performance Measures Result Through Cc And Proposed Methodology

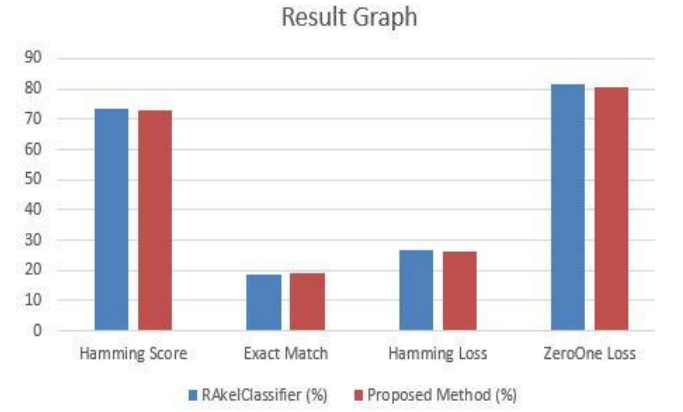
| Performance Measure | BR Classifier (%) | Proposed Method (%) |
|---|---|---|
| Hamming Score | 73 | 73 |
| Exact Match | 14.4 | 15.8 |
| Hamming Loss | 27 | 27 |
| ZeroOne Loss | 85.6 | 84.6 |
| Accuracy | 42.2 | 43.1 |




Performance Measures Result Through Rakel And Proposed Methodology

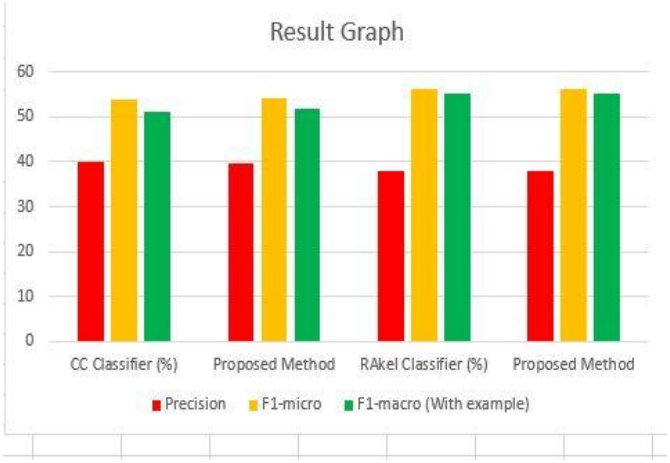
| Performance Measure | RAkel Classifier (%) | Proposed Method (%) |
|---|---|---|
| Hamming Score | 73.5 | 73.1 |
| Exact Match | 18.4 | 19.3 |
| Hamming Loss | 26.5 | 26.3 |
| ZeroOne Loss | 81.6 | 80.7 |
| Accuracy | 48.7 | 49.3 |




Experiment with Label Based Measure:
These measures are calculated for all labels by using two averaging operations, called macro-averaging and micro-averaging.

Experiment result with label measure of CC and RAkel classifier

| Performance Measure | CC Classifier (%) | Proposed Method | RAkel Classifier (%) | Proposed Method |
|---|---|---|---|---|
| Precision | 39.9 | 39.7 | 38.1 | 37.8 |
| F1-micro | 53.9 | 54.2 | 56.1 | 56.4 |
| F1-macro (With example) | 51.3 | 51.8 | 55.1 | 55.2 |




## VI. CONCLUSION

In times where data or big data is labeled as the new natural resource of the century, the importance of data mining and according techniques is ever growing. The furious development of technology enables us to collect and store massive sized and complex data sets. Therefore, real world time series data sets can take a size up to a trillion observations and even more. The overall goal is to detect new information that is hidden in these massive data sets. This experiment gives an overview of the challenges of large time series and the proposed problem solving approaches from time series data mining community. In this paper, a multi-label classifier with attribute selection algorithm for improve the accuracy and decreasing the hamming loss value for better performance has been proposed for the time series databases. The classifier receives a set of characteristics that describe the database as input and returns the set of most suitable distance measures from a set of candidates. The positive results obtained in the experimentation for various multi-label classification performance measures demonstrate that this tool is useful to simplify the attribute selection process, crucial to the time series database classification task. An important by-product of this work is the introduction of the labeling process introduced. We believe that, a method of this type has not been proposed before.

## VII. LIMITATIONS AND FUTURE WORK

Future research direction is to include new distance measures in the proposed framework. In this line, a more extensive study could be performed introducing new features that would describe other aspects of the time series databases that have not been considered in this paper. For this purpose, some of the features presented in could be considered. Another proposal for future work includes an optimization of the temporal costs associated with the calculation of the characteristics. Some of the features introduced in this study, such as the shift, are computationally quite expensive to calculate, which could be an inconvenience when working with particularly large databases. Since only means, medians, standard deviations and other general statistics are calculated, strategies such as sampling the time series database could be applied to reduce this computational cost. In the same line, reducing the number of parameters

associated to the characteristics could also improve the applicability of the proposal. Finally, some insights into the definition of the parameters of the distance measures have been included through- out the paper, but no extended experimentation has been carried out on this topic. Studying the relationship between the characteristics of the databases and the parameters that define each distance could be useful to simplify the selection of a distance measure even more.

REFERENCES

[1] Usue Mori, Alexander Mendiburu, and Jose A. Lozano, "Similarity Measure Selection for Clustering Time Series Databases" IEEE Transactions on Knowledge And Data Engineering, Vol. 28, No. 1, January 2016

[2] J. A. Ryan. (2013). Quantmod: Quantitative financial modelling framework. r package version 0.4-0 [Online]. Available: http:// CRAN.Rproject.org /package= quantmod.

[3] Hong, Y.Y.; Wu, C.P. Day-ahead electricity price forecasting using a hybrid principal component analysis network. Energies 2012, 5, 4711–4725.

[4] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," Data Mining Knowl.Discovery, vol. 26, no. 2, pp. 275–309, Feb. 2012.

[5] W. Constantine and D. Percival. (2012). WMTSA: Wavelet methods for time series analysis [Online]. Available: http://cran.r-project.org/package=wmtsa.

[6] P. Esling and C. Agon, "Time-series data mining," ACM Comput. Surveys, vol. 45, no. 1, pp. 1–34, Nov. 2012.

[7] K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths, "Comparison of correlation analysis techniques for irregularly sampled time series," Nonlinear Processes Geophysics, vol. 18, no. 3, pp. 389–404,Jun. 2011.

[8] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," Data Mining Knowl. Discovery, vol. 13, no. 3, pp. 335–364, May 2006.

[9] Liao, T.W. Clustering of time series data—A survey. Pattern Recognit. 2005, 38, 1857–1874.

[10] T. K € ohler and D. Lorenz. (2005). A comparison of denoising methods for one dimensional time series. Tech. Rep. [Online]. Available: http://www.math.uni-bremen.de/zetem/DFGSchwerpunkt/preprints/orig/lore nz20051dreport.pdf

[11] Zervas, G.; Ruger, S. The Curse of Dimensionality and Document Clustering. In Proceedings of 1999 IEE Colloquium on Microengineering in Optics and Optoelectronics (Ref. No. 1999/187), London, UK, 16 November 1999; pp. 19:1–19:3.

[12] Rodgers, J.L.; Nicewander, W.A. Thirteen ways to look at the correlation coefficient. Am. Stat.988, 42, 59–66.

[13] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A Seasonal-trend decomposition procedure based on loess," J. Official Statist., vol. 6, no. 1, pp. 3–73, 1990.