Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015
ISSN 2074-8523

Classifying Weather Time Series Using Feature-based Approach

Shakirah Mohd Taib[1], Azuraliza Abu Bakar[2], Abdul Razak Hamdan[2], and Sharifah Mastura Syed Abdullah[3]

[1]Department of Computer and Information Sciences
Universiti Teknologi Petronas,32610 Bandar Seri Iskandar, Perak
e-mail:shakita@petronas.com.my
[2]Center of Artificial Intelligence
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia,43600 Bandar Baru Bangi
e-mail:azuraliza@ukm.edu.my, arh@ukm.edu.my
[3]Institute of Climate Change
Universiti Kebangsaan Malaysia,43600 Bandar Baru Bangi
e-mail:mastura@ukm.edu.my

Abstract

Classification of weather time series is beneficial for weather forecasting problem. The classification can assist in identifying weather patterns for certain periods. In addition, extracting patterns from weather time series provides useful insights about the weather conditions to the domain experts. In this paper, we present the classification of weather time series using feature based approach that extracts feature vectors from the time series and performs the classification based on local and global features. The experimental results show that feature-based method with random forest performs well with more number of subsequences and may achieve comparable results with other methods.

Keywords: *Time Series Classifcation, Feature-based Method, Weather Time Series, Random Forest.*

1 Introduction

Weather time series are historical data obtained at regular intervals to predict future patterns based on the observation of past patterns. The weather observation includes rainfall, temperature and wind. Monitoring the trends and patterns changes in weather time series is important as it may be incorporated in other applications from other domains such as transportation, public health and business marketing. The two major data mining tasks in the analysis of weather time series patterns are classification and clustering. The algorithms for time series classification can be developed using two approaches; feature-based and instance-based approaches [1]. Instance-based approach uses similarity measures between training and test dataset to predict time series while feature-based approach classifies time series based on the features vectors extracted from training dataset. Feature-based method is widely used in image classification. It classifies image based on extracted features that describes the color, edge and texture of the images as well as the intensity and brightness of the image pixels. In time series classification, feature vector can consist of mean, skewness and Fourier transform coefficient.

2 Instance-based Method

An instance-based method involves measuring the distance between new time series and the time series in training dataset [2]. The common method widely used in time series classification is one Nearest Neighbor (1NN) classifier. The method predicts the test data by searching for similar training data points. The similarity of the data is defined using similarity measures such as Euclidean distance and Dynamic Time Warping (DTW) distance. The simplest measure is Euclidean distance which estimates dissimilarity between two time series using the following equation:

$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

Where n is the length of the time series, and x and y are the i-th element of time series x and y, respectively. The larger the difference between two series means they are less similar and they are completely similar if the distance is zero. However, we cannot measure the difference between two series with different length as it calculate distance of each pair of points. DTW is often used to solve this problem by calculating the minimum distance recursively using the following formula:

$$DTW(i,j) = (x_i, y_j) + \min\{D_{i,j-1}, D_{i-1,j,} D_{i-1,j-1}\} \quad (2)$$

Where i =1,...M and j =1,...N. M and N is the length of the time series x and y respectively.

NN with Euclidean distance is efficient in terms of time and space but it is weak in classification accuracy. However if the NN employs DTW to calculate the distance between time series, there will be limited information into the classifier and it will be not easily interpreted even though the time series is segmented based on DTW [1]. The problem can be solved by identifying the subsequences of time series using Euclidean distance or using kernel based approach with Support Vector Machine (SVM) classifier as an alternative method to classify time series.

3 Feature-based method

The feature-based method requires a method to extract feature vectors from a set of time series. A new time series is predicted based on the model built from the extracted features. There are two types of features namely global and local features. Global features such as mean or variance are compact representation while local features are extracted from time series segments. Local patterns support the global features by providing patterns that define the class of the time series and the relation between patterns may be important in the classification process. Features can be extracted using genetic algorithm [3], [4], multilayer neural network [5], interval of time series [6], Fourier Transform co-efficient [7], [8], [9] and statistical methods. The features including measures of trend, seasonality, periodicity, serial correlation, skewness, kurtosis, chaos, nonlinearity, and self-similarity are extracted to represent time series [2]. Table 1 shows the feature-based methods and their extracted features for time series classification.

Table 1: Feature-based methods for time series classification

Feature-based method	Extracted features	Studies
Genetic algorithm	Patterns	[3], [4]
Neural Network	Statistical values	[5], [10], [11]
Fourier Transform	Fourier coefficients	[1], [8],[9], [12]
Coefficient (DWT,DCT, S-Transform)	Statistical values, Coefficients, Segmentations	
Piecewise Linear Approximation (PLA)	Segmentations	[13]

Kernel Method	Segmentations	[14]
Interval-based method	Interval-based literals	[6]

Generally this method is faster than instance-based method but the selection of extraction and classification algorithm is important because the performance of feature-based approach is dependent on the efficiency of the classification algorithms [15]. Even though this method has an advantage on computational time, the global features may omit some important characteristic due to the use of standard classification algorithm.

3.1 Feature Vector Representation

A feature vector is a vector of numerical features that represent objects such as image and text. The feature values that represent image might relate to the pixel while features that represent text are the term occurrence frequencies. These numerical representations can be processed and analyzed using statistical analysis. Feature vectors are often combined with weights to determine a score for a prediction task. There are many studies on feature vector representation for time series in various domains. Gupta et al. [16] have presented a wavelet-based time series technique to extract feature vector for the prediction of protein structural class. The feature vectors that summarize the variance information of different biological properties of amino acids were extracted from mapped amino acid sequence. Their proposed approach showed a better accuracy compared to the existing approaches for protein classifcation. Rahman et al. [17] developed a feature vector representation based on rainfall time series to predict the closure of shellfish farm. They have combined Fourier Transformation Feature with the k-NN classifier and found that this combination performs better than other method for their case study.

4 A Bag-of-Features to Cluster Time Series

Bag-of-features (BOF) is a process where the complex objects are characterized by feature vectors of subobjects. BOF representation is a popular method in computer vision area for content-based image retrieval, natural scene classification, and object detection. The concept of representation is based on bag-of-words which is widely used in document classification. This representation is simple but can give a very good performance. It allows the integration of local information from segments of time series in an efficient way.

A time series bag-of-features (TSBF) that proposed by Baydogan [1] considers fixed

and variable length intervals and includes shape-based features such as slope and variance. This method provides a different representation by learning the BOF representation through a classifier on the interval features. Multiple uniform subsequences are extracted from each time series from random locations with random lengths. The subsequences are systematically segmented and features such as mean, variance and standard deviation properties are calculated to measures properties from different locations and dilation. The patterns can be represented over shorter segments. Therefore the subsequences are partitioned into intervals. A lower bound of the subsequence is set as a proportion of the time series length using parameter

$$z\ (0 < z \quad 1) \tag{3}$$

The minimum interval length is also defined to ensure meaningful features are extracted. The extracted features are recorded in a matrix and are calculated to construct a codebook. A supervised process is used to construct a codebook and a learner by adding location information. Linear regression models are fit on the intervals to extract the slope of the fitted regression line. A random forest (RF) classifier is used to generate class probability estimates for codebooks as well as to classify the time series in the datasets.

4.1 Random Forest

Random forest is an ensemble learning method developed by Breiman [18] for classification by constructing a multitude of decision trees at training time. It grows many classification trees and each tree gives a classification that known as the votes for that class. Then a new object is classified based on the highest votes from the tree in the forest. Adele Cutler has developed an algorithm to induce a random forest by combining bagging idea and random selection of features [19] and applied it to many domains such as microarray and ecology classification. The test error of bagged model can be estimated without cross-validation. About two-thirds of the observation is used by each bagged tree and the remaining one-third of the observations is referred as the out-of-bag (OOB) observation.

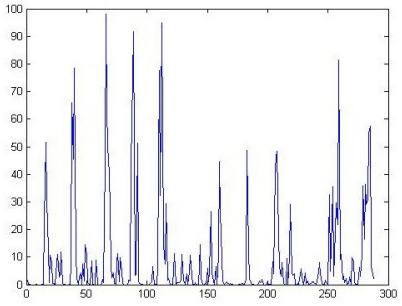
4.2 Codebook and Learning

In TSBF, the same class label for the original time series is defined for each instance. The instance is a subsequence that is extracted from each time series and all instances are trained using a supervised learner that extracts histograms of classification results to construct a codebook. The codebook summarizes the local information of the time series. TSBF also generates global features that provide information about the shape,

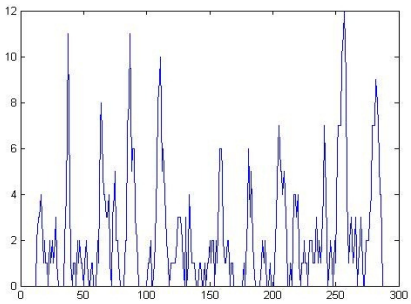
level and distribution of the values. Finally the time series are classified using the codebook and the global features. The global feature such as autocorrelation can be added to the representation to improve the result of classification . The TSBF algorithm does not require many parameters for its setting. The codebook is determined from three parameters; the minimum interval length, the number of bins and the subsequence length.

5 Weather Dataset

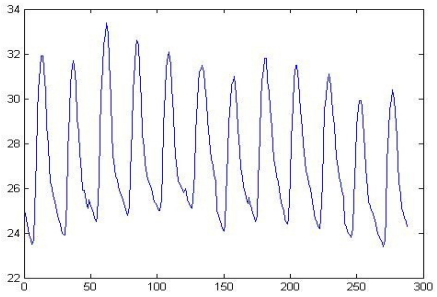
The weather dataset used for this study consists of rainfall amount, rainfall frequency and temperature time series. The length of each time series is 288 which is the total number of average value of hourly observation for each month. The data were collected at three different stations in Selangor namely Petaling Jaya, Universiti Malaya and Subang. The rainfall amount and temperature time series started from 1981 to 2008 while rainfall frequency time series data were collected from 1991 to 2008. Fig.1 shows the examples of three time series in the weather dataset recorded in 1992 at Petaling Jaya weather station.



(a) Rainfall Amount



(b) Rainfall Frequency



(c) Temperature

Fig. 1: The examples of time series for weather time series dataset

The raw data in the dataset were recorded without classification label. Before we implement the feature-based classification , the raw time series were transformed as symbolic representation and were clustered using hierarchical method.

5.1 Data Preparation -Weather Time Series Clustering

For the clustering process, the numeric time series in the weather dataset were represented as Symbolic Aggregation approXimation (SAX). SAX [20] is a symbolic representation method that is widely used to reduce the size of data without losing much information within the time series. Discretizing the time series before clustering can significantly improve the accuracy in the presence of outliers [21].

Fig.2 shows the SAX for rainfall amount time series in 1992 recorded at Petaling Jaya Station. Data was transformed into Piecewise Aggregate Approximation (PAA) representation and each representation was symbolized into a discrete string. The number of PAA segments was set to 12 and the alphabet size is 4. The alphabet are noted as a, b, c, d, e

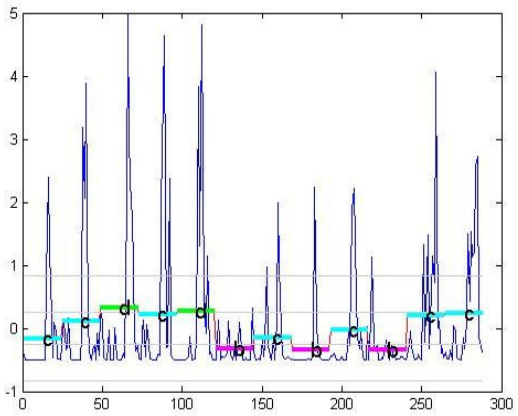


Fig. 2: SAX representation for rainfall amount recorded at Petaling Jaya station in 1992

The SAX data have been clustered based on hierarchical clustering method in WEKA data mining tool. A number of symbols combinations were created such as {bc}, {cbd}, {cdb} and {cdbe}. The proximity of the clusters is defined as the maximum of the Euclidean distance between any two points in the different clusters. This complete distance is less susceptible to noise and outliers [22]. Table 2 shows the result of hierarchical clustering on SAX weather time series data.

Table 2: Experimental Results of Weather Time Series

Cluster	Cluster 1	Cluster 2	Cluster 3
Rainfall Amount	23	15	18
Rainfall Frequency	15	18	3
Temperature	17	31	8

The clusters from the clustering results shown in Table 2 were used as the distribution of the data in classification process by labeling each class based on the respective clusters. We tested TSBF and comparison methods on a weather time series data as shown in Table 3.

Table 3: Characteristic of weather time series

Data	Min Value	Max Value	Training	Testing
Rainfall Amount	0	274	34	22
Rainfall Frequency	0	23	22	14
Temperature	22	35	34	22

Malaysia is essentially has tropical weather without extreme temperatures. Therefore the values for temperature time series are between 22 and 35. For the rainfall amount and frequency, the minimum value is 0 mostly recorded during dry season. While the maximum value of rainfall amount and frequency is 274 and 23 respectively. The values were recorded during rainy season. The training and testing data were randomly sampled based on the distribution of the data.

6 Results

The parameters setting for TSBF algorithm is simple as it is robust to the setting [1]. Table 4 shows the parameters setting in our experiment. The number of trees is set to 50-500. The codebook is determined from three parameters; minimum interval length, the number of bins and z level of subsequence length. The minimum interval is set to five time unit but the random generation scheme will allow larger interval lengths to be occurred based on the dataset characteristics.

Table 4: TSBF parameter setting

Parameter	Value
Number of trees	50-500
Number of features in each split	*Sqrt* (number of features)
Subsequence length proportion (z)	0.1, 0.35, 0.5, 0.75
Minimum interval length (*Wmin*)	5
Number of bin (*b*)	10

There are

$$r = \left\lfloor \frac{T}{W_{min}} \right\rfloor \tag{4}$$

possible intervals in the time series using the minimum interval length. Therefore there are 57 possible intervals in the weather time series data used in the experiment.

The number of intervals to represent the subsequence is determined as

$$d = \left\lfloor \frac{z \times T}{W_{min}} \right\rfloor \tag{5}$$

The value of *Wmin* in our experiment is 5. The TSBF algorithm generates *r* - *d*. Thus, at least one subsequence will be covered in every interval.

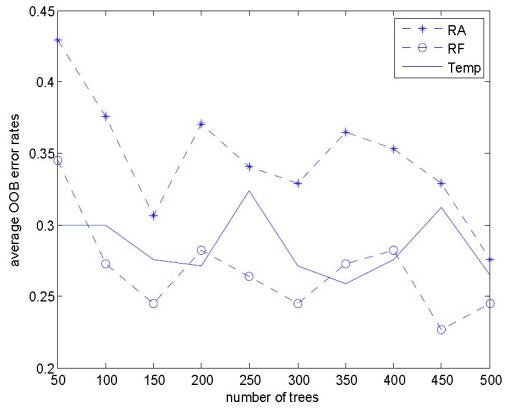
6.1 TSBF Result

Table 5 shows the average test error rates for different number of trees. The test rates for rainfall amount and temperature are decreasing but the error rates for rainfall frequency increased at 350 trees. However the latter dropped again when the number of trees is set as 500.

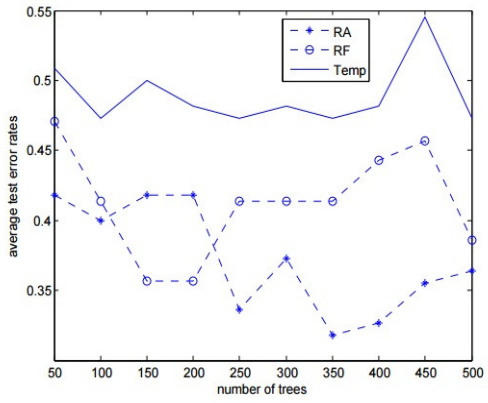
Table 5: Average Test Error Rates

Number of trees	50	150	350	500
Rainfall Amount	0.418	0.418	0.318	0.364
Rainfall Frequency	0.471	0.357	0.414	0.386
Temperature	0.509	0.500	0.473	0.473

The OOB error rates are based on training data and can be used to adjust the parameter to improve classification results in the next replication. Fig. 3 shows the OOB and test error rates for the three weather dataset.



(a) OOB error rates



(b) Test error rates

Fig. 3: OOB and test error rates over the number of trees

Both OOB and test error rates dropped at the beginning of the experiment when the number of trees is set to 100 to 500. The error rates started to fluctuate either in stable or downward trend. The average test error rates for the rainfall amount is descending over the number of trees but for rainfall amount the error rates fluctuate over the number of trees. Meanwhile the average test error rates for temperature remained stable for 150 trees until 400. However for 450 trees it increased suddenly and dropped again when we set to 500 trees. This shows the inconsistency between OOB error rate and test error rate. In order to get better accuracy, the analysis of OOB error rates with respect to different parameter settings should be conducted for each dataset.

The experiment of TSBF with 350 trees is specifically reported. The OOB error rates on training data are consistent with error rate on the test data in most of the replications except for replication 4 TSBF(z=0.1) for rainfall amount dataset and replication 1 with TSBF(z=0.5) in rainfall frequency dataset. In both cases, the OOB error rates are the lowest but they did not achieve the best test error rate. The details of the error rates are shown in Table 6-8.

Table 6: OOB and Test error rates for Rainfall Amount

TSBF	Rep 1		Rep 2		Rep 3		Rep 4		Rep 5	
	OOB	Test	OOB	Test	OOB	Test	OOB	Test	OOB	Test
z=0.1	**0.29**	0.23	**0.32**	0.36	**0.44**	0.23	**0.41**	0.50	**0.35**	0.27
z=0.25	0.50	0.36	0.53	0.46	0.53	0.46	0.50	0.46	0.68	0.41
z=0.5	0.59	0.50	0.53	0.55	0.68	0.59	0.47	0.55	0.56	0.46
z=0.75	0.71	0.59	0.65	0.60	0.77	0.41	0.56	0.59	0.65	0.41

The lowest error rate are recorded from rainfall amount test instance. The lowest error rate is 0.23 performed by TSBF(z=0.1) in the first and third repli-cations. Meanwhile the highest error rate is recorded from rainfall frequency which is 0.64 performed by TSBF(z=0.75) in the third replication. The performance of TSBF with $z = 0.1$ and $z = 0.5$ over 5 replications are illustrated in Fig. 4.

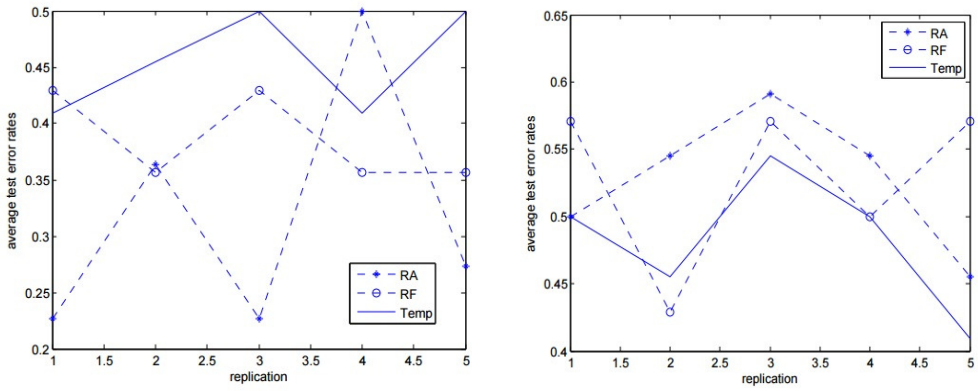
The experimental results are compared based on three different approaches, 1NN-DTW, Random Forest applied directly to the time series and TSBF as shown in Table 9. The 1NN-DTW is a notable solution for time series in many domains. The experimental results show that 1NN-DTW and Random Forest give better results for rainfall frequency time series. Random Forest classifies

Table 7: OOB and Test error rates for Rainfall Frequency

TSBF	Rep 1		Rep 2		Rep 3		Rep 4		Rep 5	
	OOB	Test	OOB	Test	OOB	Test	OOB	Test	OOB	Test
z=0.1	0.41	0.43	**0.27**	0.36	**0.23**	0.43	**0.27**	0.36	**0.27**	0.36
z=0.25	0.32	0.43	0.46	0.36	0.36	0.43	0.41	0.43	0.32	0.36
z=0.5	**0.32**	0.57	0.55	0.43	0.32	0.57	0.46	0.50	0.41	0.57
z=0.75	0.55	0.50	0.50	0.43	0.36	0.64	0.46	0.50	0.59	0.50

Table 8: OOB and Test error rates for Temperature

TSBF	Rep 1		Rep 2		Rep 3		Rep 4		Rep 5	
	OOB	Test	OOB	Test	OOB	Test	OOB	Test	OOB	Test
z=0.1	0.35	0.41	**0.21**	0.46	**0.27**	0.50	**0.27**	0.41	**0.21**	0.50
z=0.25	0.38	0.50	0.35	0.55	0.38	0.50	0.35	0.55	0.32	0.55
z=0.5	**0.35**	0.50	0.44	0.46	0.38	0.55	0.44	0.50	0.35	0.41
z=0.75	0.41	0.59	0.38	0.55	0.41	0.50	0.44	0.41	0.41	0.55



(a) TSBF (z = 0.1) (b) TSBF (z = 0.5)

Fig. 4: Average test error rates for different z values

the rainfall amount time series with the lowest error. It shows that the NN-DTW method gives poor accuracy on a larger dataset as the error rates for rainfall amount and temperature is comparable to the TSBF. On the other hand, TSBF showed poor performance on rainfall frequency and temperature time series. However, the performance of TSBF is relatively good if the z value is 0.1. The error rates are then the lowest among all the other levels.

Table 9: TSBF Test error rates

	TSBF				1NN-DTW	RForest(Raw)
	z=0.1	z=0.25	z=0.5	z=0.75		
RA	0.32	0.43	0.53	0.52	0.46	0.36
RF	0.39	0.40	0.53	0.51	0.29	0.29
Temp	0.45	0.53	0.48	0.52	0.46	0.36

Table 10 summarizes the average, minimum and maximum error rates from 5 replications of TSBF algorithm on test data. The number of tree is 350. The results show that all datasets achieve the best performance for TSBF(z =0.1). It can be concluded that the performance of TSBF on the three weather datasets improves with small z, which means the time series tested would require more number of subsequences even though the short subsquence may contains fewer features.

Table 10: TSBF Test error rates

Data	TSBF (z=0.1)			TSBF (z=0.25)			TSBF (z=0.5)			TSBF (z=0.75)		
	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
RA	0.32	0.23	0.50	0.43	0.36	0.46	0.53	0.46	0.59	0.52	0.41	0.59
RF	0.39	0.36	0.43	0.40	0.36	0.43	0.53	0.43	0.57	0.51	0.43	0.64
Temp	0.45	0.41	0.50	0.53	0.50	0.55	0.48	0.41	0.55	0.52	0.41	0.59

7 Conclusion

TSBF is a framework that was developed to learn a bag-of-feature representation for time series classification. This framework extracts random length of subsequences from random locations to allow the detection of patterns that might appear in differentlength of subsequence. The intervals in TSBF allow the detection of patterns over shorter time segments. Even though TSBF provides a comprehensive representation that handles both global and local features, the experiments using TSBF on weather time series only shows a comparable results as the 1NN-DTW method but it is slightly poorer compared to the original

Random Forest method without bag-of-features. The future direction might determine suitable threshold values of real datasets to improve the results. Further study is also to be carried out by considering other features that are specifically related to weather time series such as seasonality and trend.

ACKNOWLEDGEMENTS
The authors gratefully acknowledge the Institute of Climate Change, UKM for providing data used in this research project. This material is based on work under Grants No. ERGS/1/2012/STG07/UKM/01/1.

References

[1] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, p. 2796–2802, 2013.

[2] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, p. 1–20, 2014.

[3] D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter, and J. Theiler, "Genetic algorithms and support vector machines for time series classification," in *xyz*, 2002.

[4] A. Polanski, "Genetic algorithm search for predictive patterns in multi-dimensional time series," *Complex System*, vol. 19, no. 3, pp. 195-209, 2011.

[5] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, p. 307–319, 2003.

[6] J. J. Rodríguez, C. J. Alonso, and J. a. Maestro, "Support vector machines of interval-based features for time series classification," *Knowledge-based System*, vol. 18, p. 171–178, 2005.

[7] F. Morchen, "Time series feature extraction for data mining using dwt and dft," in *abc*, 2003.

[8] L. K. Behera, M. Nayak, and S. Mohanty, "Discrete wavelet transform and s-transform based time series data mining using multilayer perceptron neural network," *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 11, p. 8039–8046, 2011.

[9] I. Batal and M. Hauskrecht, "A supervised time series feature extraction technique using dct and dwt," in *8th International Conference on Machine Learning and Applications, ICMLA 2009* , p. 735–739, 2009.

A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based classfication of time-series data," *Information processing and Management*, vol. 56, p. 49 61, 2001.

[10] A. Kampouraki, G. Manis, and C. Nikou, "Heartbeat time series classication with support vector machines," *IEEE Transaction On Information Technology in Biomedicine*, vol. 13, no. 4, p. 512–518, 2009.

[11] W. A. Chaovalitwongse and P. M. Pardalos, "On the time series support vector machine using dynamic time warping kernel for brain activity classification," *Cybernetics and Systems Analysis*, vol. 44, no. 1, p. 125–138, 2008.

[12] N. Q. V. Hung and D. T. Anh, "Combining sax and piecewise linear approximation to improve similarity search on financial time series," in *Proceedings - 2007 International Symposium on Information Technology Convergence, ISITC 2007*, pp. 58-62, 2007.

[13] O. Kramer and F. Gieseke, "Analysis of wind energy time series with kernel methods and neural networks," in *2011 Seventh International Conference on Natural Computation*, vol. 4, p. 2381–2385, 2011.

[14] P. Geurts, "Pattern extraction for time series classification," in *Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery*, p. 115–127, " in Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery, 2001.

[15] R. Gupta, A. Mittal, and K. Singh, "A time-series-based feature extraction approach for prediction of protein structural class," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2008, 2008.

[16] A. Rahman, M. S. Shahriar, C. D'Este, G. Smith, J. McCulloch, and G. Timms, "Time-series prediction of shellfish farm closure: A comparison of alternatives," *Information Processing in Agriculture*, vol. 1, p. 5–32, Aug. 2001. L. Breiman, "Random forests," *Machine learning*, vol. 5, no. 32, pp. 1-35, 2001.

[17] A. Cutler and G. Zhao, "Pert - perfect random tree ensembles," *Computing Science and Statistics*, vol. 33, p. 490–497, 2001.

[18] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, p. 107–144, 2007.

[19] A. Bagnall and G. Janacek, "Clustering time series with clipped data," *Machine Learning*, vol. 58, no. 1, p. 151–178, 2005.

[20] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Addison Wesley, 2006