2022 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
SCIENCE, TECHNOLOGY & ENGINEERING, ARTS, MATHEMATICS & EDUCATION JUNE 7 - 9, 2022
PRINCE WAIKIKI RESORT, HONOLULU, HAWAII

TIME SERIES CLASSIFICATION USING DEEP LEARNING MODELS

MARIANI, MARIA C.
DEPARTMENT OF MATHEMATICAL SCIENCE
UNIVERSITY OF TEXAS AT EL PASO
EL PASO, USA

ARTHUR, JAMES
DEPARTMENT OF MATHEMATICAL SCIENCE
UNIVERSITY OF TEXAS AT EL PASO
EL PASO, USA

TWENEBOAH, OSEI K.
DEPARTMENT OF DATA SCIENCE
RAMAPO COLLEGE OF NEW JERSEY
MAHWAH, NEW JERSEY, USA

Time series Classification Using Deep Learning Models

1st Dr. Maria Mariani
Department of Mathematical Science
University of Texas at El Paso
El Paso, USA

2nd James Arthur
Department of Mathematical Science
University of Texas at El Paso
El Paso, USA

3rd Dr. Osei Tweneboah
Department of Data Science
Ramapo College of New Jersey
Mahwah, NJ, USA

***Abstract*—Time series classification (TSC) can be implemented with many techniques. Several of these techniques are based on analyzing 1-D signals in the time series data. We implement a new time series classification that involves a two-stage process in this work. Firstly, by using Recurrence Plots (RP) we transform the time series into 2D images. The second stage involves taking advantage of deep learning models to perform our classification. The image representation of time series introduces different feature types that are not readily available for 1D signals, and therefore our classification problem is treated as a 2D image recognition task. Experimental results show that our multistage modeling of time series is very effective compared with other traditional classification frameworks.**

***Index Terms*—Machine learning, Time Series, anomaly detection, Recurrence Plot**

I. Introduction

Time series has numerous definitions, but they all stem from the idea that a time series is a sequence of data points (measurements) with a natural temporal ordering. The time series analysis is mainly used for curve fitting, function approximation, prediction and forecasting, segmentation, classification, and clustering. Pattern creation [14] , identification [15], anomaly detection [16], and many crucial real-world pattern recognition tasks deal with time series analysis [17]. Financial data (e.g., stock market and currency exchange rates), biometrics (e.g., voice, signature, and gesture), Video processing, music mining, forecasting, industrial devices (e.g., gas sensors and laser excitation), and weather are examples of application domains with time-series nature (Nima Hatami 2019; Yann Gavet et al. 1991).

The feature types of data might systematically sort Time Series Classification (TSC) methods. Concerning the feature types, "recurrence domain" methods incorporate phantom examination and wavelet investigation, while "time series" techniques incorporate auto-relationship, auto-regression, and cross-correlation analysis. The classification strategy can likewise be partitioned into "instance-based" and "feature-based" methods. The previous measures' similarity between any incoming test sample and the training set; and assigns a name to the most comparative class (the Euclidean distance based 1-Nearest Neighbor (1-NN) and Dynamic Time Wrapping (DTW) are two well known and broadly utilized techniques for this classification. The latter first transforms the time series into the new space and extracts more discriminative and representative features used by a pattern classifier aiming at the optimum classification boundaries.

The Convolutional Neural network (CNN) is currently one of the most famous DL models. Usually, feature-based frameworks CNN do not require hand-crafted features, which makes it different from the other frameworks. Both feature learning and classification parts are bound together in one model and are advanced and mutually learned. In this manner, their performances are improved together. Multiple layers of various handling units (e.g., convolution, pooling, sigmoid/exaggerated digression squashing, rectifier, and normalization) are responsible for learning (representing) a hierarchy of features from low-level to high-level. This work attempts to investigate the performance of Recurrence Plots (RP) with the CNN model for TSC. A recurrence plot gives a method for picturing the periodic nature of any trajectory through a stage space and empowers us to research specific parts of the stage space direction through a 2D representation of the data. Because of the recent outstanding results by CNN on image classification, we first encode time-series signals as 2D plots and then treat the TSC problem as a texture recognition task. A CNN model with hidden layers followed by a fully connected layer is used.

A. Risks and Challenges

Time series applies to a unique class of problems. The approach can use data from temporal ordering to make statements about causation and focus on all change patterns over time. Time series analysis also suffers from several weaknesses. Some weaknesses are the problem of generalization from a single project, difficulty obtaining and using appropriate metrics and measures, and problems with accurately identifying the best correct model for the data. Using appropriate techniques and understanding the data to mitigate these pitfalls might help reduce or eradicate these problems.

Limitations of Time Series: Time Series is vastly applied in almost all spectrum of research. From science, statistics, and recently artificial intelligence. There are other genius ways of getting time series models where their accuracy, prediction, and cost efficiency are very reasonable. Depending on the dataset some times, some machine learning algorithms do not deal well with time.

Therefore, we aim to address the following critical research questions in this paper.

- How much improvement is achieved when we implement a new time series classification which involves a two-stage process of using Recurrence Plots (RP) to transform the time series into 2D images
- Which transformation technique, in general, is best for the given data set based on standard performance metrics?
- Is multistage modeling of time series very effective compared with other traditional classification frameworks?

The overview of this paper is as follows: Section II discusses related research on Time series classification with deep learning. Section III discusses our methodologies, presenting a brief description of the data, Recurrence Plots, and Convolution Neural Network background. Section IV spells out the results and discussions. Finally, Section VI concludes this paper and discusses future works.

II. Related works

This section briefly outlines the most recent time series forecasting and classification techniques. Researchers in [1] used numerous domains like IoT (Internet of things), signal processing, and human activity recognition. Given a time sequences dataset, the goal was to train a good model that accurately predicts a time series class. The algorithms break the problem recursively into subproblems (if applicable), store the results, and later use them when needed instead. The authors in [3] introduced a study on several past methods applied for deep learning in time series. They showed that the headway of machine learning is not difficult to distinguish the failure in machines when its significant features are chosen cautiously. Several algorithms were reviewed, and it is discussed how cost-effective it can be when well-chosen techniques are used.

The authors in [4] introduced a study on time series forest (TSF) classifier that adapts the random forest classifier to series data. It then Split the series into random intervals, with random start positions and lengths. It then extracts summary features (mean, standard deviation, and slope) into a single feature vector from each interval. It then inadvertently trains a decision tree on the extracted features. This intricate process is repeated until the desired number of trees has been built or time runs out.

Dictionary-based classifiers first transform all the real-valued time series into a sequence of succinct discrete “words.” Classification is then based on the vast distribution of the extracted symbolic words. Furthermore, researchers in [5] discussed algorithms for time series classification (TSC) that focus on capturing the frequency of pattern occurrences in a time series.

The work we present addresses the advanced technique of nonlinear data analysis and reveals several times when the phase space trajectory of the known dynamical system recurs roughly the same area in the phase space by transforming the time series into 2D images. The image representation of time series introduces different feature types that are not available for 1D signals, and therefore our classification problem is treated as a 2D image recognition task. The second stage involves taking advantage of deep learning models to perform our classification.

III. Methodology

Time series can be characterized by distinct recurrent behavior such as periodicities and unpredictable cyclicities. Furthermore, the recurrence of states is typical for dynamic nonlinear systems or stochastic processes in which time series are generated. The RP is a visualization tool that aims to investigate the phase space trajectory through a 2D representation of all recurrences. The principal idea is to reveal at which points some trajectories return to a past state, and it can be formulated as:

$$R_{i,j} = \theta_i(\epsilon_i - \|\overrightarrow{x_i} - \overrightarrow{x_i}\|), \overrightarrow{x}(.) \in R^m, i,j = 1,...,N \quad (1)$$

where N is the number of considered states $\overrightarrow{x_i}$ ϵ is a threshold distance, a norm and θ (.) the Heaviside function. The R-matrix contains textures: single dots, diagonal lines, vertical and horizontal lines, and typology information characterized as drift, homogeneous, periodic, and disrupted. Fading out to the upper left and lower right corners means the transformed data has a trend or drift.

In calculating the recurrence plot, a systematic procedure is used. First, the 2D phase space trajectory ($m = 2$) is constructed from the time series. Then, the R-matrix is derived based on the nearness of the states in the phase space. The resulting R-matrix has only $0, 1$ values that are caused by the thresholding parameter ϵ. Inspired by the different texture images obtained from the R-matrices, this paper proposed a TSC pipeline based on the CNN model. First, the raw 1D time-series signals are classically transformed into 2D recurrence texture images, and then both features and classifier are strategically learned in one unified model.

Accuracy metrics: In our research, our primary focus is the accurate prediction of the transformed 2D image, so the best measures, in this case, are other performance metrics such as the average rank and number of wins. Our primary performance metric stems from the idea that an algorithm outperforms others comparatively if it has the most significant number of wins and lowest number of average ranks. The number of 'Wins' measures counts the number of datasets that a specific algorithm obtains the lowest error rates, while Average Rank is the **mean of the error-rate ranking over entire datasets**. Aside from the RP, we compared our algorithm to other state-of-the-art algorithms like shapelets, 1-NN Dynamic time warping, Gramian Angular Field, and SAX-VSM. Our goal was to compare the RP to the other algorithms. We found it prudent to normalize the measures for each algorithm because some error rates were missing for some datasets for some algorithms.

Algorithm Recurrence Plot for transformation [1]

for **do**(i, j) with $\overrightarrow{x(i)} = \overrightarrow{x(j)}$:

$\vec{x} = \langle t_1, t_2, ..., t_T \rangle$ where two successive steps are separarted by δ and $\vec{x}(i)$ is recorded for each time step getting trajectory $X = \langle \vec{x}(t_1), \vec{x}(t_2), ..., \vec{x}(t_T) \rangle$

A 2D plot is created where the x-axis and y-axis both report $\vec{w}$, forming a $T \times T$ lattice of little squares, each with a side measuring δ

The data $\mathbf{X}$ are used to compute a matrix R formed by binary elements recording the recurrence/non-recurrence of values $\vec{x}$through the binary function:

$$\mathbf{R}(i, j) = \begin{cases} 1 & \|\vec{x}(i) - \vec{x}(j)\| \le \epsilon \\ 0 & otherwise, \end{cases}$$

where i,j $\in$ t$_1, t_2, ...t_T$

The recurrence plot then visualises $\mathbf{R}$ with a black little square of the lattice at coordinates(i,j) if $\mathbf{R}(i, j) = 1$, and a white little square if $\mathbf{R}(i, j) = 0$

end for

Output the the transformed data.

A. Recurrence Plot

The visual appearance of all recurrence plots gives details about the system's dynamics. Caused by the characteristic behavior of the phase space trajectory, a recurrence plot contains typical small-scale structures, such as single dots, diagonal lines, and vertical/horizontal lines (or a mixture of the latter, which combines into extended clusters). The large-scale structure, also called texture, can be visually characterized by homogeneous, periodic drift, or disrupted. For example, the plot can show that if the trajectory is strictly periodic with period T, then all such pairs of times will be separated by a multiple of T and visible as diagonal lines. (Wikipedia 2020). For the transformation, a single, isolated recurrence plot can occur if states are rare, do not persist for any time, or fluctuate heavily. However, they are not a unique sign of chance or noise (for example, in maps). When the data has periodic/ quasi-periodic patterns, it means there are cyclicities in the process; the time distance between periodic patterns (e.g., lines) corresponds to the period; long diagonal lines with different distances to each other reveal a quasi-periodic process. Furthermore, if there are vertical and horizontal lines/clusters, it means the data do not change or change slowly for some time, an indication for laminar states. Fig. 1 shows a set of classical time series and their corresponding transformed data.

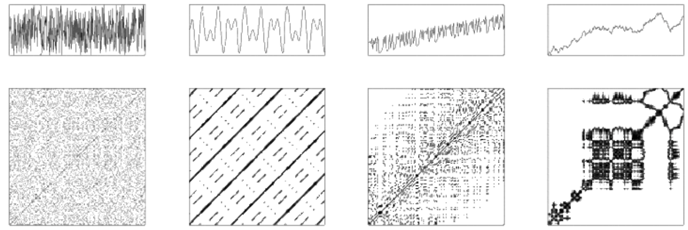

Fig. 1: From left uncorrelated stochastic data (white noise), harmonic oscillation with two frequencies, chaotic data (logistic map) with linear trend, and data.

Recurrence plot from paradigmatic systems gives an excellent introduction to characteristic typology and texture. Moreover, their quantification offers a better objective way to investigate the considered system. From Fig. 2, we realize there is a fading to the upper left and lower right corners. This means the Olive Oil data is nonstationarity; the process contains a trend or drift.

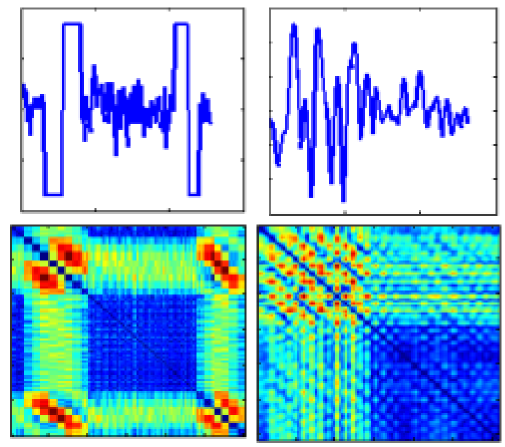

Fig. 2: Time series to image encoding on two different datasets from the UCR archive: FaceAll, OliveOil (from left to right, respectively).

IV. Data background

The data used were from the University of East Anglia and the University of California Riverside (UEA /UCR) Time Series Classification Repository. It is comprised of several different time-series data. Some data do not allow for unequal length series, so the unequal length problems are all padded with missing values. The UCR Archive currently contains 128 datasets. Fifteen of these are unequal lengths, and one (Fungi) has a single instance per class in the train files. To have a thorough and fair experimental evaluation of all approaches, we tested each algorithm on the whole UCR/UEA archive (Chen et al. 2015b; Bagnall et al. 2017), which contains 85 univariate time-series datasets. The datasets possess varying characteristics, such as the length of the series and class.

A. Structure of the data:

The choice of validating on the UCR/UEA archive is motivated by having datasets from almost all different domains, which have been broken down into seven different categories (Motion Capture, Spectrographs, ECG, Image Outline, Sensor

Readings, Electric Devices, and Simulated Data). The process of classification of a time series with deep learning is very systematic. First, the time series is transformed into image data with the help of the recurrence plot. After that, the transformed image is used as a bedrock for the convolutional neural network, as in Fig. 3.

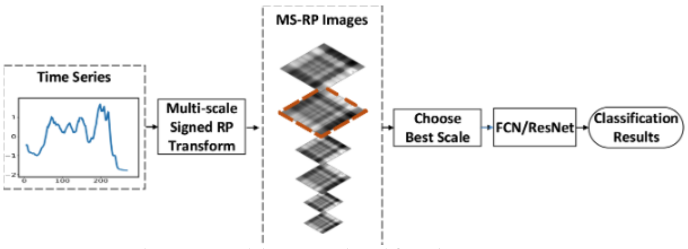

Fig. 3: Multistage classification process.

B. *CNN Architecture*

The CNN is trained by feeding it with images of size 28 × 28 × 4 (width, height, channels). Each image corresponds to an activity instance, and the four channels are the distance matrices for the magnitude and x, y, and z axes, as outlined in the previous section. The activity labels are also passed to the CNN at training time as one-hot encoded vectors. Then the Input layer, passes the data to two consecutive convectional layers, after which max pooling and dropout ($p = 0.25$) are applied. Next, the data passes through two more convolutional layers, and again, max pooling and dropout ($p = 0.25$) are applied. After that, the data is flattened and passed through a fully connected layer of 512 units with dropout ($p = 0.50$).

Each convolutional layer uses a kernel of size 3 with a stride size of 1. The max-pooling layers use a pool size of 2. The kernels for the first two convolutional layers were set to 16 and 32 for the last two. Finally, a fully connected layer with six units (for the six activities) and a softmax activation function is used to produce the final output, i.e., the probability for each activity.

V. Results and discussion

A. *Accuracy and Error Metrics*

Evaluating a deep learning algorithm is an essential part of any research. Accuracy is used to measure the performance of models; however, it is not enough to truly judge models using accuracy alone. Using the average rank and the number of wins helps to confirm the efficiency of the models. The average rank and number of wins help identify the best algorithm for the exact project. It also enforces that due diligence must be done before any algorithm is chosen to model any data. Almost all the state-of-the-art algorithms had good performances, but the RP was the best amongst them. From Table 1, It had the highest number of wins and the smallest average rank. RP was the best because of the informative features that it possesses. RP takes advantage of CNN's high performance on image classification; time-series signals are first transformed into texture images (using RP) and then handled by a deep CNN model.

TABLE I: The Table of Error Performance Rate

Dataset	**Class**	**RP**	**GAF**	**SAX-VSM**	**1-NN-DTW**	**Shapelet**
50 Words	50	0.26	0.30	**0**	0.31	0.44
Adiac	37	**0.28**	0.37	0.38	0.39	0.51
Beef	5	**0.08**	0.23	0.033	0.36	0.44
CBF	3	0.005	0.009	0.02	**0.003**	0.05
ECG200	2	**0**	**0**	**0**	**0**	0.06
FaceAll	2	**0**	0.09	0.14	0.23	0.22
Face4	14	**0.19**	0.23	0.20	**0.19**	0.40
Fish	4	**0**	0.06	0.007	0.17	0.09
Gunpoint	7	0.085	0.114	**0.007**	0.17	0.19
Lightning	2	**0**	0.08	0.017	0.093	0.061
Average Rank		**2.15**	3.40	3.0	4.08	5.75
Number of Wins		**7**	1	3	3	0

The error rate is given in Table 1. Other time series classification algorithms were compared to the proposed algorithm. Shapelets, Gramian Angular Field (GAF), SAX-VSM, and 1-NN DTW. The table comprises the summary of the performance of the error results of all the considered algorithms. The better the algorithm, the smaller the average rank. It was noted that SAX-VSM had the second-best accurate algorithm with an average rank of 3 and several wins of 3. This was followed closely by GAF, with an average rank of 3.40 with one win. 1-NN-DTW and Shapelets had the least accuracy, with an average rank of 4.08 and 5.75, respectively. RP had the best accuracy with an average rank of 2.15 and 7 wins. It is noticed that for a TSC, RP can give image transforms of the CNN model and is more efficient than the other traditional frameworks.

VI. Conclusion and Future work

CNN used in image processing can be two-dimensional (2D)or 1D. They can as well be used for time series classification because time series have an intense time locality that convolutions can extract. Since multivariate time series have the same 2D dimensional data structures as images, CNN images are very suitable for handling multivariate time series.CNN can learn a different level of time series features in a classified state.RP can give a good visualization of the m- dimensional phase space trajectory, giving the best performance in the experiment. In future work, different algorithms like inception time and Echo state Networks are to be considered because they can speed up the training process since they are sparsely connected with most of their weights fixed. In conclusion, high accuracy and potential high scalability make the perfect algorithm for most deep learning projects.

References

[1] Nikos Kafritsas, Time series classification using Dynamic Time Warping, 2021.

[2] Hassan Ismail Fawaz, Germain Forestier,Jonathan Weber,Lhassane Idoumghar,Pierre-Alain Muller, "A review: Deep learning for time series classification",2019

[3] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir, "A Time Series Forest for Classification and Feature Extraction," 2013.

[4] Zeki Murat Çınar et al., "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0," MDPI, 2020.

[5] M. De Benedetti et al, "Anomaly detection and predictive maintenance for photovoltaic systems", Neurocomputing 310 (2018) 59–68; Volume: 310, pp 59-68 (16 May 2018).

[6] Prashanth Pillai et al., "A Hybrid Approach for Fusing Physics and Data for Failure Prediction," International Journal of Prognostics and Health Management, 2016.

[7] Hastie T., Tibshirani, R., Friedman, J.(2009), The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edition, Springer,2008.

[8] Harada, Yoshiyuki Yamagata, Yoriyuki Mizuno, Osamu Choi, Eunhye. (2017). Log-based Anomaly Detection of CPS Using a Statistical Method.

[9] Kohonen, Teuvo Honkela, Timo. (2007). Kohonen Network. Scholarpedia. 2. 1568. 10.4249/scholarpedia.1568.

[10] P. Konar, P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs)," Applied Soft Computing 11 (2011) 4203–4211; Volume: 11, Issue: 6, pp 4203-4211 (March 25, 2011)

[11] M. De Benedetti et al, "Anomaly detection and predictive maintenance for photovoltaic systems", Neurocomputing 310 (2018) 59–68; Volume: 310, pp 59-68 (16 May 2018)

[12] Breiman, L. (2001). Random Forests. Machine Learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324

[13] Eischeid, Jon Pasteris, Phillip Diaz, Henry Plantico, Marc Lott, Neal. (2000). Creating a Serially Complete, National Daily Time Series of Temperature and Precipitation for the Western United States.

[14] Jessica Lin, Sheri Williamson, Kirk Borne, David Debarr. (2000). Pattern Recognition in Time Series Series. https://cs.gmu.edu/ jessica/publications/astronomy11.pdf

[15] Chen, S., Min, W., Chen, R. (2013). MODEL IDENTIFICATION FOR TIME SERIES WITH DEPENDENT INNOVATIONS. Statistica Sinica, 23(2), 873–899. http://www.jstor.org/stable/24310366

[16] Aditya Bhattacharya (2020). Effective Approaches for Time Series Anomaly Detection https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1

[17] Carolina Bento. (2020)Time-Series Forecasting in Real Life: Budget forecasting with ARIMA https://towardsdatascience.com/time-series-forecasting-in-real-life-budget-forecasting-with-arima-d5ec57e634cb

[18] Nima Hatami,Yann Gavet and Johan Debayle . (2000). Classification of Time-Series Images Using Deep Convolutional Neural Networks