2022-12-01

Time Series Classification With Multistage Modeling Using Deep Learning

James Arthur
University of Texas at El Paso

TIME SERIES CLASSIFICATION WITH MULTISTAGE MODELING USING DEEP LEARNING

JAMES EKOW ARTHUR

Master's Program in Computational Science

APPROVED:

Maria Christina Mariani, Ph.D., Chair

Osei Tweneboah, Ph.D., Co-Chair

Joe Guthrie, Ph.D.

Thompson Sarkodie-Gyan, Ph.D.

Suneel Chatla, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

to my

Mother,

with love

TIME SERIES CLASSIFICATION WITH MULTISTAGE MODELING USING DEEP LEARNING

by

JAMES EKOW ARTHUR

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Computational Science Program

THE UNIVERSITY OF TEXAS AT EL PASO

December 2022

Acknowledgements

I want to express my deep-felt gratitude to my advisor, Dr. Maria Christina Mariani of the Mathematical and Data Science Department at The University of Texas at El Paso, for her advice, encouragement, enduring patience, and constant support. She never ceased believing in me, always providing clear explanations when I was lost and constantly motivating and supporting me.

And also, my sincerest appreciation is to my co-advisor, Dr. Osei Tweneboah of the Department of Data Science at the Ramapo College of New Jersey. His technical advice, guidance, explanation, and patience helped me in this work.

Additionally, I want to thank The University of Texas at El Paso Computational Science Department professors and staff for all their hard work and dedication, providing the means to complete my degree and prepare for a career as a computational scientist.

Abstract

Time series classification (TSC) can be efficiently implemented with several techniques. Many techniques are based on analyzing 1-D signals in the time series data. In this work, we make an intrinsic analytical implementation of a new time series classification that involves a two-stage process. Firstly, by using Recurrence Plots (RP), we transform the time series into 2D images. The second stage consists in taking advantage of deep learning models to perform our classification. The image illustration of time series introduces different feature types that are not available for all 1D signals, and therefore our classification problem is treated as a 2D image recognition task. Experimental results show that our multistage time series modeling is exceptionally effective compared with an alternate traditional classification framework. A significant amount of data is stored in the form of time series. Climatic measurements, performances, medical tests, stock exchanges, satellite locations, and political opinions are all data saved as a time series. Time series data can be any information collected successively in time. Since processes are often measured relative to time, this data type exists in almost every task. Some examples of it are stock prices, industrial processes, electronic health records, human activities, sensor readings, and language. Because it is ubiquitous, extracting value from time series data around us is only practical. Time series classification has an extensive range of applications. The novel application of time series can start from the identification of Soil sedimentation, stock market anomalies, and the Spread of viruses to automated detection of heart and brain diseases. Time series classification can be evaluated or conducted with many techniques. Most of these techniques have two stages. The first approach uses mathematical methods, statistics, or programming tools to represent time series as feature vectors. Secondly, an algorithm can measure the difference between the time series one wants to classify. When one uses some algorithm to classify data, one can implement anything from k-nearest neighbors and SVMs to deep neural network models.

The convolution Neural Network (CNN) model also has many impeccable attributes. It jointly and automatically allows different learning levels of representations with a classifier. Therefore, Recurrence Plots RP and CNN in a compact, a unified framework is expected to boost the time series classification recognition rate. Experimental results on the UCR time-series classification demonstrate the rigid competitive accuracy of the proposed approach compared to the existing deep architectures and the state-of-the-art TSC algorithms.

Table of Contents

Chapter 1

Introduction

The definitions of time series can be enormous. However, they all emanate from the concept that time series is a sequence of real data points (measurements) with a natural temporal ordering. Pattern creation, identification, anomaly detection, and many important pattern recognition tasks holistically deal with time series analysis. Time series analysis is mainly grouped into curve fitting, functional approximations, prediction and forecasting, segmentation, classification, and clustering. In a univariate time-series classification, $X_n \beta Y_n$ is defined so that n^{th} series of length $l : X_n = (X_{n1}, X_{n2}, ..X_{nn})$ is associated with a class label $y_n \equiv 1, 2, .., c$. Time-series classification problems can be easily adapted to other tasks, such as anomaly detection and clustering. The feature types of data might systematically sort Time-Series Classification (TSC) methods. Regarding feature types, "Recurrence Plot" methods incorporate phantom examination and wavelet investigation, while "time series" techniques incorporate auto-relationship, auto-regression, and cross-correlation analysis. The classification strategy can be partitioned into "instance-based" and "feature-based" methods. The previous measures similarity between any incoming test sample signal and the training set; and assigns a name to the most comparative class (the Euclidean distance-based k-Nearest Neighbor (k-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. Deep Learning models have, over the years, achieved a high recognition rate for computer vision and speech recognition. The Convolutional Neural Networks (CNN) is one of the most well-known DL models. While other traditional" feature-based" classification framework needs this, CNN does not require any handcrafted features. The feature learning and classification parts are unified into one model and are systematically learned jointly. This makes their performances mutually enhanced. This work investigates the performance of Recurrence Plots (RP) inside the deep CNN model for TSC. RP gives a manner to visualize the systematic nature of a trajectory through a phase area and permits us to research specific components of the m-dimensional phase area through a 2D representation. Because of the recent thorough results by CNN on image recognitions, we first encode time series signals as 2D plots and then deal with the TSC problem as a texture recognition task. A CNN model with two hidden layers followed by a fully connected layer is used.

1.1 Background Of Study

Significant results have been achieved by processing most data with deep learning techniques, especially convolutional neural networks (CNN). (Antonio Fernandez 2015). CNN's performances in reading, processing, wrangling, and extracting essential features of two-dimensional data have contributed to its popularity for data classification. However, even in scenarios where input data are not formatted as an image, many transformation methods have helped to apply CNNs to data types. Time series data is a data structure that can be modeled well to solve problems from computer vision and image processing using deep learning.

Spectrograms are one of the most famous representations for signals, in which time series send information with time and frequency as magnitude and dimensions. Though spectrograms are known to be graphical representations of frequency spectrum over time, some delicate nuances exist between these representations and visual aids, like pictures taken with a camera or paintings. However, in spectrograms, all local relationships are represented using different domains. They have non-local relationships, in contrast to pictures. Furthermore, this concept complicates the local feature extraction feeding 2D

CNN layers with spectrograms. We can use this information to take advantage of 2D CNN considering visual representations with inherent spatial invariance as they efficiently provide the best input for a convolutional layer.

Recurrence plots are an advanced technique for visually representing multivariate nonlinear data. In essence, this refers to a graph illustrating a matrix, where elements correspond to the time the data recurs to a particular state or phase. Recurrent behavior, such as periodicities or irregular cyclicities, is a fundamental property of natural deterministic dynamical systems, like nonlinear or chaotic systems.

1.2 Problem Statement

During the past couple of years, there have been conscious efforts to introduce computers to the world. Data analysis, wrangling, acquisition, interpretation, and forecasting have been prevalent in the world using computers. The availability of temporal data has over the years increased significantly. Many applications are based on time series, so efficient algorithms must be proposed and used. Many of the algorithms available require some feature engineering as a separate work before the classification is done. This means there is a loss of information and increased development time. Deep Learning solves this problem. Deep learning models already incorporate feature engineering internally, optimizing it and eliminating its need. Therefore they can extract information from the time series faster, more direct, and more completely.

Thousands of data are compiled and used for many reasons. However, time series classification has become one of the most challenging data mining in data. Many classification problems can be assiduously treated as a Time Series classification problem. Hence a more accessible and less cost-effective classification approach to help solve cumbersome problems is critical. Time series is very present in health care, finance, and cybersecurity. Interest has been shown in other fields about time series. The main goal of a classification algorithm is to build a model that associates with an object's probability of belonging to the possible classes according to the features of the objects related to each class. To be more specific. A time series classification problem is a classification problem where the objects of the datasets are univariate or multivariate time series. There are techniques implemented to enable a smooth classification process with CNN. The Gramian Angular field and Markov Transition fields are two techniques that can be well used to convert time series data into image data. However, the Recurrence plot technique gives a better and less cost-effective approach to transforming the time series data into image data and for CNN implementation. RP provides a systematic way to visualize a trajectory through a phase area and an avenue to investigate specific aspects of the m-dimensional phase area trajectory through a detailed 2D representation. We first encode time series signals as 2D plots and then treat the TSC problem as a texture recognition task. A CNN model with two hidden layers followed by a fully connected layer is used.

1.3 Objective of the study

The objective of the study includes the following;

- Converting Time Series data into image Data using Recurrence Plot
- To introduce Convolution Neural Networks for classification
- Proof that Neural Networks can compute any function
- Is there a simple algorithm for computer intelligence?
- Show other approaches to Deep Neural networks

The desire to solve a particular case of the general problem of determining the accuracy of such indirect measurements is the motivation for the work in this thesis.

1.4 Methodology

Time series can be characterized by distinct recurrent behavior, such as periodic and irregular cyclicities. The RP is a visualization tool that aims to explore all the m- dimensional phase space trajectories through a detailed 2D representation of its recurrences. The main idea is to reveal at which points any trajectories return to a previous state and if it can be formulated as:

$$R_{i,j} = \theta_i(\epsilon_i - \mid \overrightarrow{x_i} - \overrightarrow{x_i} \mid), \overrightarrow{x}(.) \in R^m, i, j = 1, ..., N \tag{1.1}$$

Where N is the number of considered states $\overrightarrow{x_i}$ ϵ is a threshold distance, a norm, and $\theta(.)$ the Heaviside function. The R-matrix contains both textures, all single dots, diagonal strains of lines, vertical and horizontal lines, and typology data characterized as homogeneous, periodic, and drift. For instance, a fade to the top upper left and lower bottom right corners means that the systematic process contains a trend or drift. Moreover, vertical and horizontal lines/clusters show that some states will not change or change slowly concerning time, and this can be interpreted as a laminar state,

In calculating the recurrence plot, a systematic procedure is used.

- Firstly, a 2D phase space trajectory ($m = 2$) is built from the time series.
- Then, the R-matrix is derived based on the closeness of the states in the phase area. The resulting R-matrix has only $0, 1$ values, caused by thresholding parameter ϵ. Inspired by the unique texture images derived from the R-matrices, this paper proposes a TSC pipeline based on the CNN model. Firstly, the raw 1D time-series signals are transformed into 2D recurrence images. And then, both features and classifier are learned in one unified model.

The proposed 2-stage CNN architecture for TSC is then started. Recurrence plot images are resized to $28 * 28$, $56 * 56$, or $64 * 64$ (depending on the data) and fed into the CNN model.

1.4.1 CNN Architecture used

A 2-stage deep CNN model is applied with a 1-channel input of size 28×28. Each feature learning stage represents a different feature level and comprises convolution (filter), pooling activation, and pooling operators. The input and output of each of the layers are known as feature maps. The convolutional layer is the primary building block of a CNN and exploits spatially local correlation by re-enforcing a local connectivity pattern between neurons of all adjacent layers. The activation function introduces non-linearity into the networks and permits them to learn the complex models. We then applied ReLU (Rectified Linear Units) because it systematically trains the neural networks faster without a significant penalty to generalization accuracy. Subsampling reduces input resolution and makes it robust to all slight variations for previously learned features. It combines the outputs of the i-1th layer into a single input in the ith layer over a range of local neighborhoods. When the feature extraction is done, the feature maps are flattened and fed into a fully functional connected layer for classification. The fully connected layers connect every neuron in one layer to another. This is the same principle as the traditional multi-layer perceptron (MLP). The proposed pipeline for TSC.

1.4.2 Learning Procedure

Training the above CNN architecture is almost similar to the Machine Learning problems. The gradient-based optimization method (error back-propagation algorithm) estimates the model's parameters. Stochastic gradient descent (SGD) is used for faster convergence to update the parameters considered. The training phase has two main classical steps: Propagation and gradual weight update. Each propagation stage involves a feed-forward and error back-propagation passes. The former determines the feature maps on the input vector by passing from one layer to another layer until reaching the output. The latter calculates the propagation errors with respect to the loss function for the desired predicted output. The predicted error on each layer is used for calculating the derivatives by using the chain rule of the derivative. Once the derivatives of parameters are obtained, the weight is gradually updated. The output delta and input activation are then multiplied to derive the weight gradient. And then, a ratio of the weight's gradient considered is subtracted from the weight. This cycle is repeated over and over until the network reaches a satisfactory validation error.

1.5 Significance of the Study

This study is a step towards understanding and appreciating the usage of artificial intelligence, particularly machine learning, concerning Deep Learning. A less costly and efficient technique to solve classification in time series is critical. The methods, algorithms, and processes made mentioned in the study can be implemented in the study of

- Electrocardiogram analysis. (records are saved in time series form. Distinguishing a disease is a TSC problem)
- Gesture recognition. (Devices record a series of images to interpret the user's gestures. And also, identifying a correct gesture is a TSC problem)
- Anomaly Detection (This is the identification of unusual events. Often, the data for anomaly detection are time series data. Recognizing an anomaly is a TSC problem)

1.6 Organization of Thesis

This study is organized into five chapters and outlined as follows

- **Introduction** : This chapter presents a general introduction to the study with a background to the study, the problem statement, objectives, Methodology, and the significance of the study.
- **Literature Review**: In this chapter, various works of literature about Time series Classification techniques are presented.
- **Methodology**: This chapter discusses various methods adopted for the study and the algorithm used.
- **Model Formulation and Classification Analysis**: In this chapter, the model is formulated for each technique, and the analysis of each technique is explained.
- **Simulations and Deep Learning information**: In this chapter, further analyses are discussed with simulations.
- **Conclusion and Recommendations**: This chapter concludes the entire study and lays out some recommendations for future studies.

Chapter 2

Literature Review

In this chapter, we briefly introduce machine learning and the techniques used in this study. Machine Learning (ML) is an integral branch of Artificial Intelligence. It is a system that takes in data, tries to find patterns, trains itself using it, and brings out an outcome.
ML algorithms have critical advantages over human professionals. Firstly, machines can work faster than humans. Calculating spatial problems may usually take a long time. A computer can do thousands of iterations and calculations in seconds. Machines can repeat themselves thousands of times without getting exhausted. Humans do it, too. We call it practice. While practice may make perfect, more practice can only put a human being even close to the computational speed of a computer. Another advantage is the excellent accuracy of machines.
With the recent availability of the Internet of Things technology, there is so much data out in the world that humans cannot possibly go through. They can do work faster than humans. If different algorithms can perform the same task, one is right to question which algorithm is better. For example, if two programs are made based on two different algorithms to find the smallest number in an unordered list. For the same list of unordered numbers (or same set of input) and on the same machine, one measure of efficiency can be the speed or quickness of the program, and another can be minimum memory usage. Thus, time and space are the usual measures to test the efficiency of an algorithm. In some situations, time and space can be interrelated; that is, the reduction in memory usage leads to fast execution of the algorithm. For example, an efficient algorithm enabling a program to handle complete input data in cache memory will allow faster program execution.

2.0.1 Deep Learning

Deep Learning is one of the main subsets of machine learning, which can easily be explained as a neural network with three or more layers. Neural networks try to emulate the behavior of the human brains enabling systems to cluster data and make reasonable predictions with high accuracy (IBM Cloud Education 2020). Deep learning is the gradual scaling up of neural network data structures. It is the optimal solution when working with large-scale datasets. Even though a neural network with a single layer can create impressive approximate predictions, more hidden layers can help optimize and refine for much better accuracy.

Deep Learning helps most artificial intelligence (AI) applications and services and renders an improvement in automation, performing analytical and physical tasks. There is a Deep learning technology behind most products and services.

2.0.2 Deep Learning vs. Machine Learning

Deep Learning is a known subset of machine learning, but it differs from each other. Deep Learning is different from machine learning due to the type of data it works with and the methods it learns with.

All machine learning algorithms leverage well-structured, labeled data to make predictions. It generally uses classical pre-processing to organize the data into a structured format. The specific features are defined from the input data for the said model and organized into well-defined tables.

Deep Learning eliminates some of the data pre-processing typically involved with all machine learning. These algorithms can take data and process unstructured data, like text, images, and videos, and it automates all feature extractions, removing some of the sole dependency on experts.

Most phones with facial recognition work well. The technology can determine the person's features and compare them to the scanned face on the phone. Deep learning algorithms can determine which main features (e.g., ears) are most significant to distinguish each animal from another. A human expert establishes this hierarchy of features manually. (IBM Cloud Education 2020)

2.0.3 Applications of Machine Learning

Machine learning has done right by responding to numerous real-world challenges, yet there are various issues for which AI advancement is required.

- Automatic Recognition of Handwritten Postal Codes How machine learning has helped the post office can well appreciate. People write differently and have peculiar ways of writing some alphabets. Hence deducing the word that was written is sometimes challenging to read. A handwritten recognition algorithm was the way to go.

- Language identification The ability to identify a particular language is sometimes vital. When security companies want to transcribe or interpret languages, a machine learning algorithm comes in handy.

2.0.4 Applications of Deep Learning

A few years ago, we never anticipated deep learning applications to bring us self-driving cars and virtual assistants like Siri, Alexa, and Google Assistant. However, today, these creations are part of our everyday life. Deep Learning fascinates us with its endless possibilities, such as fraud detection, anomaly detection, and pixel restoration. Deep Learning is an ever-growing industry; upskilling rapidly with the help of an accessible introduction to deep learning course can help you understand the basic concepts and power ahead of your career. (Marina Chatterjee; 2022)

- Self-Driving Cars

Deep Learning is the concept that brings autonomous driving to life. Thousands of data are fed to a system to create a model, then train the machines learn and test the results. The Uber Labs at Pittsburg is not only functioning with the idea of making driverless cars humdrum but also integrating numerous clever capabilities and features, including food delivery options using driverless cars. The predominant difficulty for self-driving automobile builders is coping with unprecedented scenarios. (Marina Chatterjee; 2022)

- News Aggregation and Fraud News Detection

  There is a way to filter out all the bad news from the news feed. Rigorous use of deep Learning in news aggregation bolsters efforts to customize news per reader. While this may not be new, newer levels of sophistication to understand, define and monitor reader personas are being met to screen out news as per social, geographical, and economic parameters, along with the individual preferences of most readers. Fake news detection, on the other hand, is an essential asset in today's world. The internet has become the hub of all genuine and phony information. It becomes tough to distinguish between fake news as bots replicate it across channels automatically. Deep Learning helps to develop classifiers that can detect all fake or biased news, remove it from your feed, and warn you of potential privacy breaches. Training and validating a deep-learning neural network for news detection is challenging. The data is plagued with opinions and sentiments, and no party can decide whether the news is neutral or biased. (Marina Chatterjee; 2022)

- Visual Recognition

  When you imagine yourself going through a plethora of old images, you decide to get a few framed, but first, you would like to sort them out. The manual effort was the only way to accomplish this without metadata. The maximum you could do was briefly sort them out based on dates, then location. However, downloaded images lack that metadata sometimes. Deep learning can do this work with ease.

Searching for a particular photo from a library (let us say a dataset as large as Google's picture library) requires state-of-the-art visual recognition systems consisting of several layers from basic to advance to recognize elements. Large-scale image Visual recognition through deep neural networks boosts growth in this digital media management segment by extensively using convolutional neural networks, TensorFlow, and Python. (Marina Chatterjee; 2022)

2.1 Mathematics Behind Deep Learning

Deep learning aims to scale machine learning to the challenges needed to solve artificial intelligence. This means being able to understand high-dimensional data with a structure. We want AI algorithms to understand raw images, representing speech, documents containing multiple words and punctuation characters, and dataset evolution.

2.1.1 Density Estimation

For an input x, machine learning deduces an estimate of the actual density(x) under the data-generating distribution. This requires a single output, but it also requires a complete detailed comprehension of the entire input. If an element of the vector is unusual, the dynamic system must assign it a very low probability.

2.1.2 Denoising

For a damaged or incorrectly observed input x, the ML system returns an estimate of the initial or correct x. For example, the machine learning system may be processed to ask to remove dust or scratches from data or a picture.

2.1.3 Missing Value Imputation

Given the dataset of some elements of x, a model is asked to output estimates of or a probability distribution over some elements of x. This requires multiple outputs. The model could be demanded to restore any of the elements of x so that it should understand the entire input.

2.1.4 Sampling

The model generates new samples from the distribution p(x). Applications include speech synthesis, producing new waveforms that sound like natural human speech. This requires multiple output values and a good model of the entire input. If the samples have even one element drawn from the wrong distribution, then the sampling process is wrong

2.1.5 Runtime concerning the cost of inference

Suppose we want to experiment and deduce an inference task where we use our model of the joint distribution P(x) to compute some other distribution, such as the marginal distribution P(x1) or the conditional distributions($x_2|x_2$). Computing these distributions will require summing across the entire table, so the runtime of these operations is as high as the intractable memory cost of storing the model.

2.1.6 Runtime concerning the cost of Sampling

suppose we want to draw a sample from the model. The naive way to do this is to sample some value $u \epsilon U(0, 1)$, then iterate through the table, adding up the probability values until they exceed and return the outcome corresponding to that position in the table. This requires reading through the whole table in the worst case, so it has the exact exponential cost as the other operations.

2.1.7 Directed Graph

Formally, a directed graphical model defined on variables x is defined by a directed acyclic graph G whose vertices are the random variables in the model, and set of local conditional probability distributions $P(x_i|PaG(x_i))$, where aG(xi) gives the parents offering. The probability distribution over is given $by(x) = \prod ip(x_i|PaG(xi))$.

Chapter 3

Time Series Classification with Convolutional Neural Network

This chapter will introduce time series, recurrence plots, and convolutional neural network implementation.

3.1 Time Series

A time series is a sequence of raw data points occurring in successive order over time. This may be contrasted with cross-sectional data, which captures a point in time. (Adam Hayes 2021). A critical type of time series is a stationary time series. A time series is stated to be strictly stationary if its properties are not affected by any change in the time origin. That is, if the joint probability distribution of the observations $y_t, y_{t+1}, ...y_{t+n}$ is the same as the joint probability distribution of the observations $y_{t+k}, y_{t+k+1}, y_{t+k+2}, ...y_{t+k+n}$ then the time series is strictly stationary. When $n = 0$, the stationarity assumption means that the probability distribution of $y_r$is the same for all periods and can be written as $f(Y)$.

3.1.1 Important factors to consider for time series models

- Accuracy and development time.
  The best model is not the model that is the fastest. The best model is a model with high accuracy and optimum development time. Some models may give faster outputs, but the accuracy levels are bad.

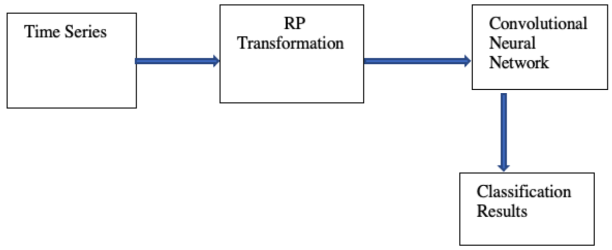

Figure 3.1: The chaining process of the DL classification.

- What-if scenerio

  A good model should be very viable. Performing what-if scenario analysis and tracking potential forecast accuracy should be easy. When the metrics involved or the datasets are changed, they should be able to work if the data structure is the same.

- Ease of use

  Re-training and data pipeline setup saves time and makes the generation of predictions effortless.

3.1.2 Time Series Analysis

Time series analysis is the collection of data at specific intervals to identify trends, cycles, and seasonal variances to aid in forecasting a future event.
Trends are consecutive increases or decreases in measurement over time. A trend could last several days, months, or years. A trend will reverse in almost every observation during the measurement lifetime. This reversal is sometimes referred to as a correction. Corrections occur in the economy, the stock market, and business. It normally follows unprecedented growth or loss. Seasonal variances are measured over several months and are associated with a specific time of the year. Retailers realize seasonal growth in sales during November and December. For the rest of the year, sales are relatively flat. The year can be divided into four quarters. The first three quarters show a small sales volume, giving way to a significant growth in sales in the fourth quarter.

3.2 Recurrence Plot

Recurrence analysis is based on the repeatability of time series states and allows presenting a time series as a geometric structure. The topology of such geométric structures will enable us to reveal and analyze the characteristic features of time series dynamics of different natures. The work aims to classify ECG time series based on the construction of recurrence plots. After transforming the time series into recurrence plots, two approaches are applied for classification: using quantitative recurrence characteristics as the classifier features and the recognition of recurrence plot Images using a convolutional neural network.

3.2.1 Recurrence Plots interpretation

The recurrence plot is a square matrix $RP_{i,j}, i, j = 1, ...N$ The element of $RP_{i,j}$ is equal to 1. If the distance between points $x(t_i)$ and $x(t_j)$in phase space does not exceed some predetermined value ϵ, When the opposite case occurs , $RP_{i,j}$ is equal to 0

A recurrence plot (RP) is an advanced technique for nonlinear data analysis. It is a visualization (or a graph) of a square matrix, whereby the matrix elements correspond to those times in which a state of a classical dynamical system recurs (columns and rows correspond then to a particular pair of times). Technically, the Recurrence plot shows when the phase space area trajectory of the classical dynamical system enters roughly the same area.

3.3 Recurrent Plot Definitions

Recurrence Rate is the density of recurrent points.

$$RR = \frac{1}{N^2} \sum_{i.j=1}^{N} RP^{\epsilon}_{i,j} \tag{3.1}$$

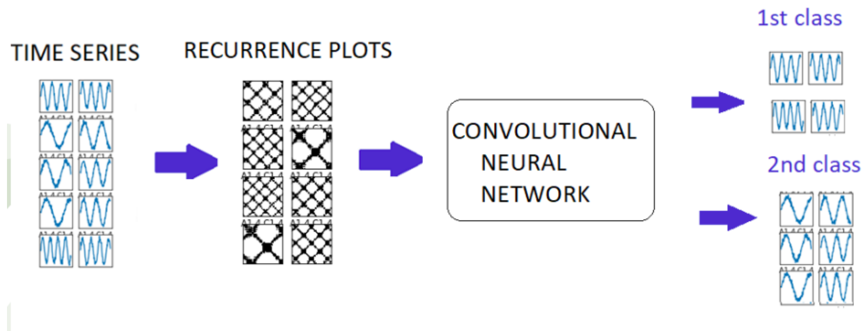

Figure 3.2: Recurrence plot chain process

Probabilty P_τ that the system recurs to the ϵ neighborhood of point x_i of the trajectory after τ time steps.

$$P_\tau = \frac{1}{N-\tau}\sum_{i.j=1}^{N}\phi(\epsilon- \mid x_i - x_{i+1} \mid) \tag{3.2}$$

Diagonal Lines $P(l) = l_i; i = 1, ..., N_i$ is the frequency distribution of the lengths of the diagonal lines, where the l_i is the length of the i-th diagonal line, $N_i$is the number of diagonal lines.

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, 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 multiple 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 irregular patterns (e.g., lines) corresponds to the period; long

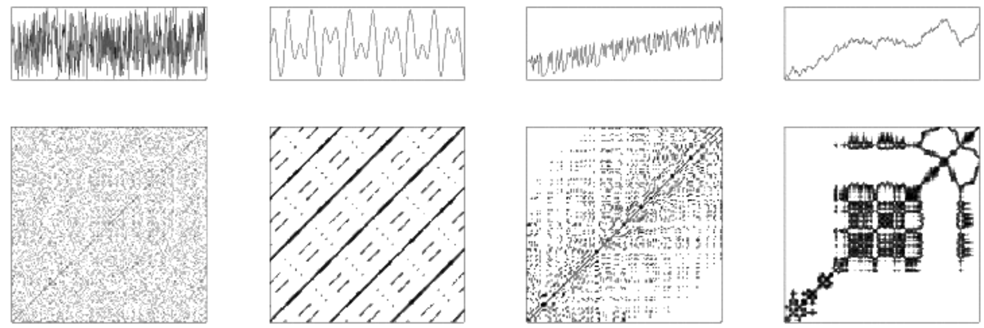

Figure 3.3: From left uncorrelated stochastic data (white noise), harmonic oscillation with two frequencies, chaotic data (logistic map) with linear trend, and data.

diagonal lines with different distances to each other reveal a quasi-periodic process. Furthermore, if there are vertical and horizontal lines/clusters, the data stay mostly the same for some time, indicating laminar states. Fig 3 shows a set of classical time series and their corresponding transformed 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. Fig 3.4 shows a fading to the upper left and lower right corners. This means that the Olive Oil data is nonstationary; the process contains a trend or drift.

3.4 Classifier for Images

For image classification, convolutional neural networks are the best. The CNN contains five layers; the first two are convolutional. The designed output of the last layer is fed to a two-sided softmax, producing a distribution over two classes. Neurons in fully well-connected layers are connected to all of the neurons in the previous layers. The activation function used is the ReLu. The method of batch normalization was used in layer regularization.

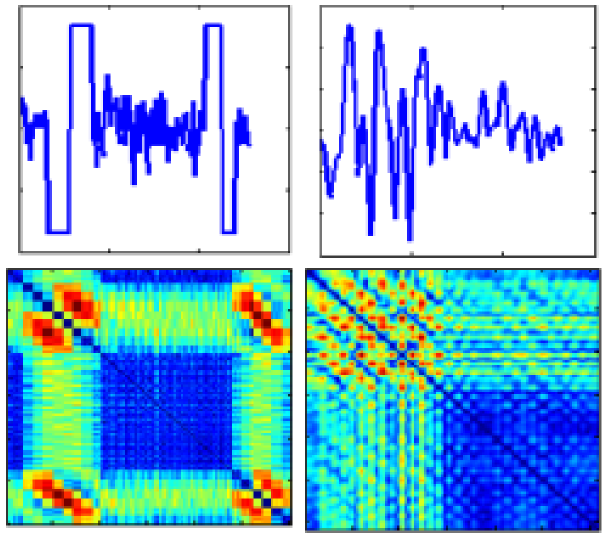

Figure 3.4: Time series to image encoding on two datasets from the UCR archive: FaceAll and OliveOil(from left to right, respectively).

3.5 ReLu Function

The rectified linear unit provides a straightforward nonlinear transformation. Given an element x, the function is defined as the maximum of that element and 0:

$$ReLu(x) = max(x, 0) \tag{3.3}$$

The RelU function retains only positive elements and discards all negative elements by setting the corresponding activations to 0. When the input is negative, the RELU function is 0, and also when the input is positive, the derivative of the RELu function is 1

3.5.1 Activation Functions

Activation functions decide whether a neuron should be well activated by calculating the weighted sum and adding bias to it. They are differentiable operators to transform input signals into outputs, while most add non-linearity. Because activation functions are fundamental to deep learning, let's briefly survey some common activation functions.

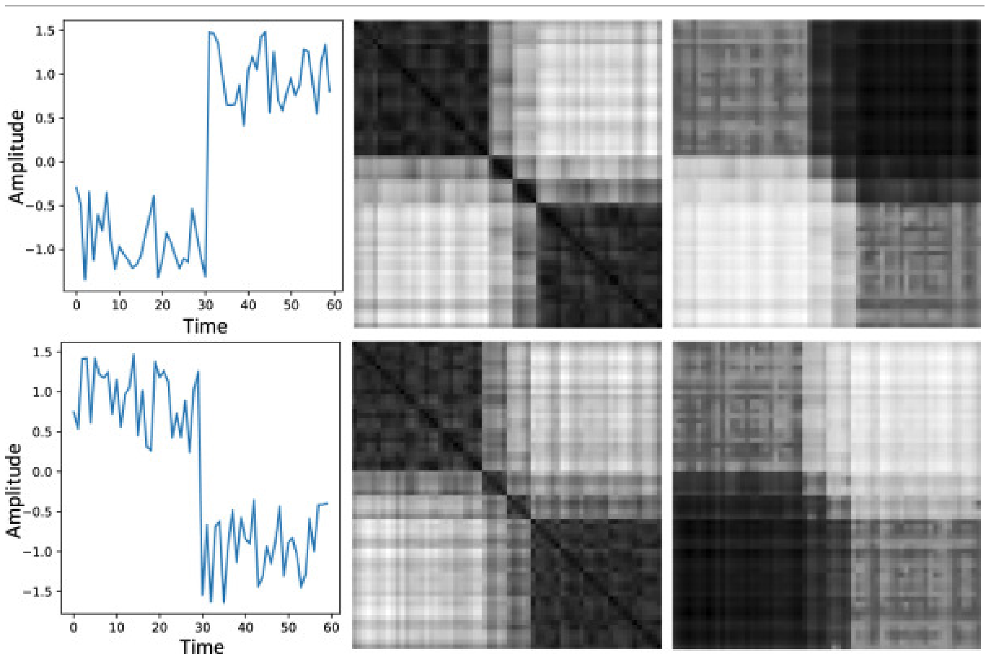

Figure 3.5: The RP images of two sequences with opposite tendencies. (left column: two sequences of the 'SyntheticControl' dataset, middle column: the original RP images of these sequences, right column: the signed RP images of these sequences

3.6 Asymmetric RP for encoding long time series

Information will be lost when the large-size recurrence plot images of long sequences are directly reduced to smaller sizes. To address this information loss, a new asymmetric recurrence plot is needed. The original one will be symmetric along the leading diagonal, causing information redundancy. Hence we have to divide the long sequence into two pieces and encode each as a recurrence plot. The two extracted matrices are reassembled into an asymmetric image.

3.6.1 Introducing convolutional networks

A Convolutional Neural Network is a DL algorithm that takes as input an image or a multivariate time series and can successfully capture all the spatial and temporal patterns by

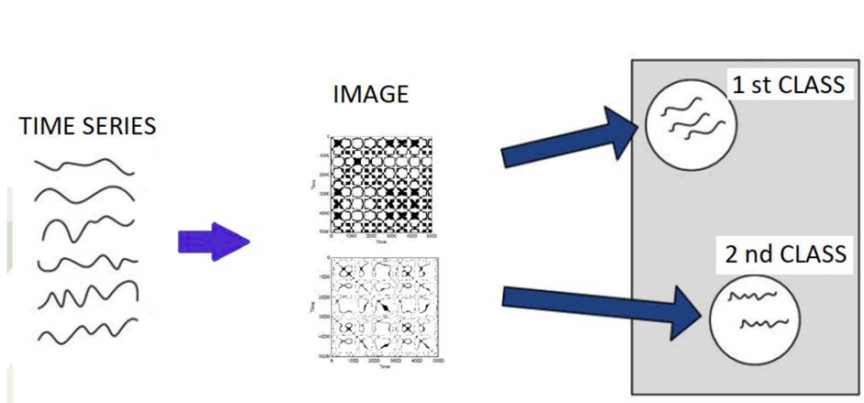

Figure 3.6: Recurrence plot feature selection for convolutional neural network

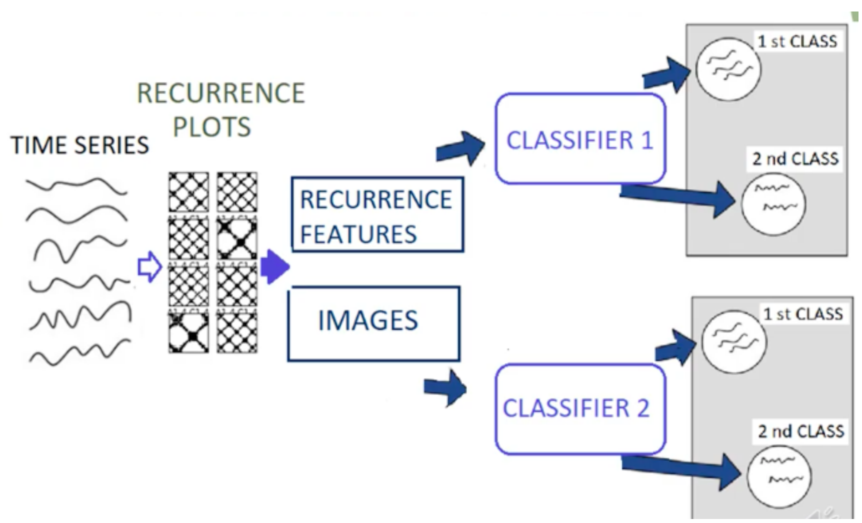

Figure 3.7: Time series to Recurrence plot chain process for classification

applying good, trainable filters and assigning importance to these patterns using trainable weights. The pre-processing required in a Convolutional Neural Network is much lower than in other classification algorithms. Even though many filter methods are hand-engineered, Convolutional Neural Networks can learn these filters.

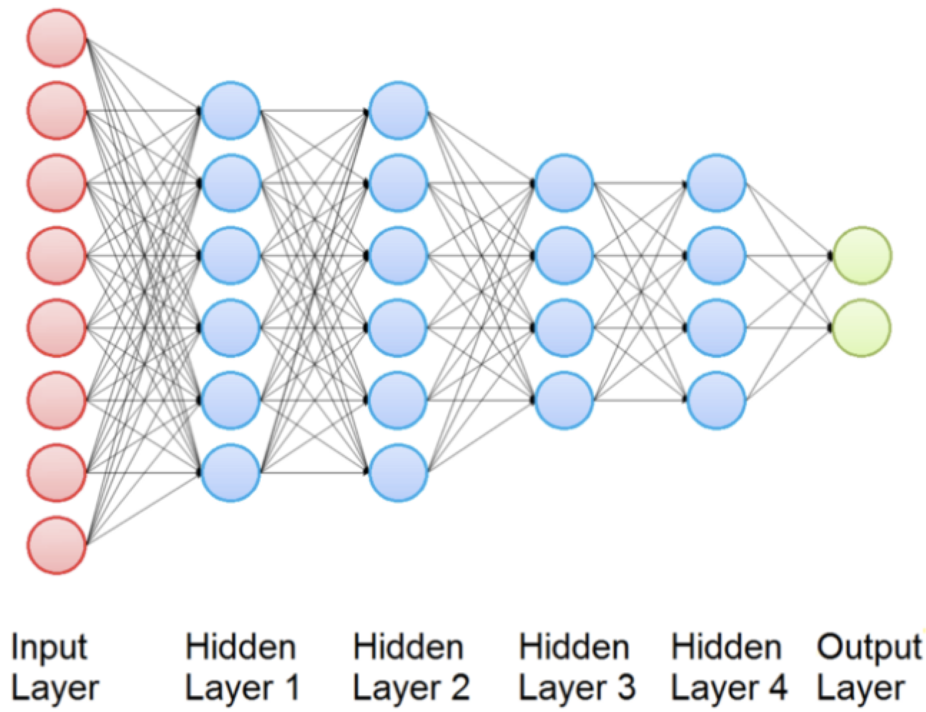

Figure 3.8: A classical convolution Neural Network with layers

When well implemented, deep learning helps improve model accuracy and performance. The techniques can be used in a variety of architectures and are not model-specific:

Examples include:

- Batch normalization
- Latent variables
- Activation functions

We can see that a Convolutional Neural Network is composed of three different layers: They are the Convolutional layer, the Pooling layer, and the Fully-connected Layer

- **Convolutional Layer** The main principle of the 2D convolution is to drag a convolution kernel on the image. The convolution between the kernel and part of the images is treated at all positions. The kernel then moves by the number of pixels (called stride). When the stride is relatively small, it becomes redundant information. It is called zero padding in this case. If the size of the image input is W x H x C, the volume of the output becomes $W_0 * H_0 * C_0$

$$W_0 = \frac{W_i - b + 2p}{s} + 1 \tag{3.4}$$

$$H_0 = \frac{H_i - b + 2p}{s} + 1, \tag{3.5}$$

If the image has three channels and if $b_l (l = 1, ..., C_0)$ denote 5 * 5 * 3 kernels (where three corresponds to the number of channels of the input image), the convolution with the image I with the kernel b_l corresponding to the formula.

$$b_l * I(i, j) = \sum_{c=0}^{2} \sum_{n=0}^{4} \sum_{m=0}^{4} b_l(n, m, c) I(i + n - 2, i + m - 2, c) \tag{3.6}$$

The convolution operations are combined with an activation function ϕ(generally the Relu activation function): if we consider a kernel K of size $b * b$, if x is a $b * b$ patch of the image, the activation is obtained by sliding the $b * b$ window and computing $z(x) = \phi(b * x + bs)$, where bs is a bias.

- **Pooling Layer** The Pooling layers allow taking the mean or the maximum on all patches of images. It acts on small patches of the image. When a $2x2$ patch is considered, we will take the maximum value to define the output layer a stride $s = 2$ will be divided by the height and width of the image. It is possible to reduce the dimension. This can be done by taking a stride larger than one without padding. An advantage of pooling is that it makes the network less sensitive to small translations of the input images. reduce the dimension by

- **Fully-Connected Layer** After considering several convolutions and pooling layers, The Convolution Neural Network ends with several fully connected layers. The tensor

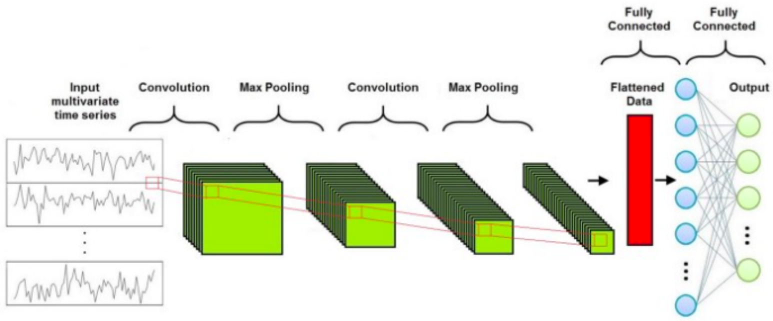

Figure 3.9: Deep Learning System

that we have at the output of the layers is transformed into a vector, and perceptron layers are added.

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 as one-hot encoded vectors at training time. Then the Input layer passes the data to two consecutive convectional layers, after which max pooling and dropout (p = 0.25) are used. 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 five hundred and twelve units with dropout (p = 0.50).

Each convolutional layer uses a kernel of size three 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.

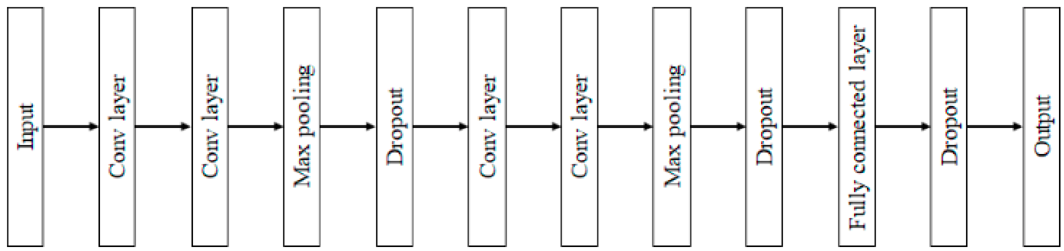

Figure 3.10: CNN Architecture

3.6.2 Softmax and Cross Entropy Loss

To minimize the difference between o and the labels y. While it turns out that treating classification as a vector-valued regression problem efficiently works surprisingly well, it is, however, lacking in the following ways:

- There is no guarantee that the outputs sum up to 1 in the way we expect probabilities to behave.
- There is no guarantee that the outputs oi are even nonnegative, even if their outputs sum up to 1, or that they do not exceed 1.

3.7 Can CNN compute all Functions

It is possible to get results with just a single hidden layer. However, can we replace networks with external, single-hidden-layer networks? While, in principle, that is possible, there are good practical reasons to use deep networks. Deep networks have a very hierarchical structure, making them well adapted to learning the hierarchies of all knowledge that seem helpful in solving problems. When approaching issues such as image processing or recognition, it helps to use a classical system that understands not just individual pixelations but also increasingly more complicated concepts, from edges to more geometric shapes, through complex, multi-object scenes. The universality of the neural network shows that neural networks can compute any function. Empirical evidence also suggests that deep networks are best adapted to learn the functions that help solve many real-world problems.

Chapter 4

Multistage Modeling Analysis and Results

4.1 Structure of the data:

The choice of validating on the UCR/UEA archive is motivated by having datasets from almost all domains broken down into seven 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.

4.2 Analysis of The Data

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.

4.2.1 Classification Nuances

The benchmark approaches for time series arrangement could be sorted into three gatherings: distance-based, include-based, and brain network based. . We are interested in the classification conduct of various models since they all perform similarly on the same dataset with different accuracies and their feature areas and learn classifiers be noted.

4.3 Classification Transforms

4.3.1 Gramian Angular Fields

Gramian Angular Fields (GAF) are images representing a time series in a non-Cartesian coordinates system (i.e., each point on the plane is referenced by an X and Y axis). Instead, the coordinates are mapped by a Polar Ordinate system (i.e., each point on the plane is determined by a distance from a reference point and an angle from a reference direction). Thus each GAF represents a temporal correlation between each time point. (Johann Faouzi et al)

4.3.2 Markov Transition matrix

A time series $X = (X_1, ..., X_n)$of real-valued observations is discretized based on its quantile bins; that is, each bin is assigned to its corresponding bin Q_j with $j \epsilon 1, ..., Q$ and Q is the number of quantile bins, resulting into a discretize-valued time series of length n. Considering this discretized-valued time series as observations of a first-order Markov chain, one can compute the number of occurrences of pairs of back-to-back bins for every pair of bins, resulting in a Q $\times$ Q matrix.

The Markov transition matrix is insensitive to the temporal distribution of the time series X since it only captures the frequencies of the transition but not at which time points they occurred. Moreover, its size depends on the number of bins and not the length of the time series, although larger time series may allow for a larger number of bins. To

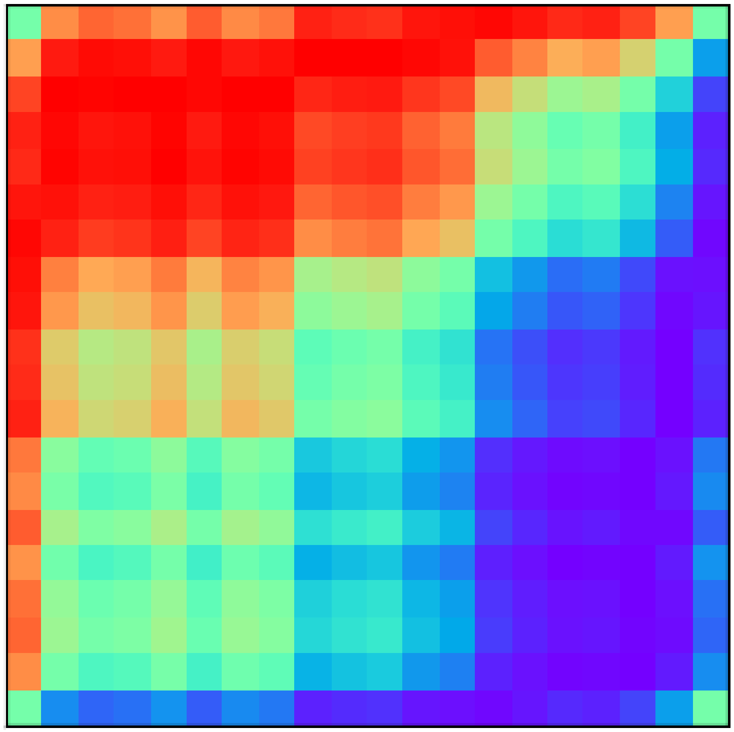

Figure 4.1: Gramian Angular Field output on Oliveoil data considered. This is the image generated when the time series dataset is transformed into an image

overcome these issues, the Markov transition matrix is projected onto an n × n matrix called the Markov transition field. (Johann Faouzi et al)

4.4 Residual Neural Network

A residual neural Network is an artificial neural network with a very deep forward neural network. It has many layers with much dense neural networks than most ANN. A non-residual network is called a plain network. With Resnet, all layers are expanded and stay closer to the manifolds, therefore learning faster. This makes it very susceptible to perturbations that cause it to leave the manifold and needs more training data to recover. (Wikipedia; 2022)

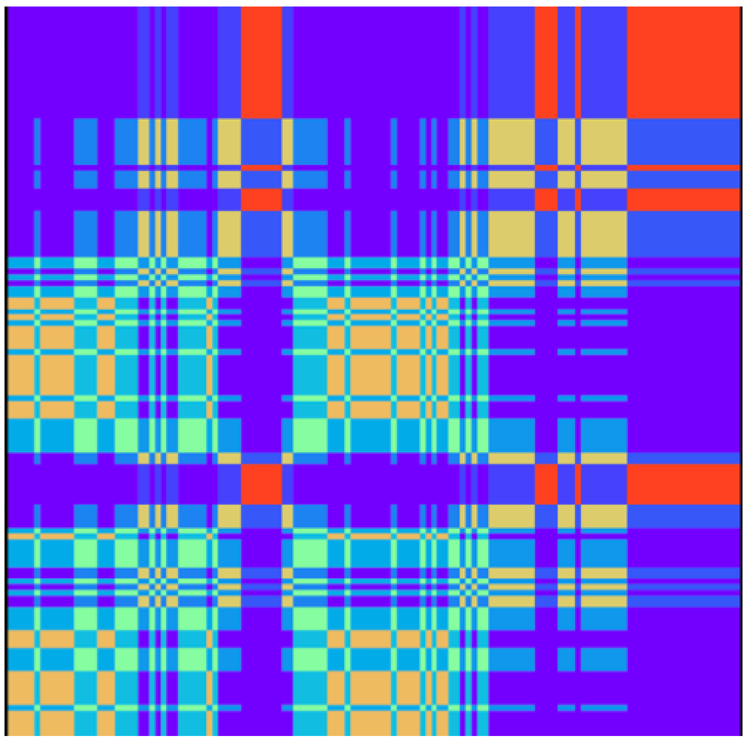

Figure 4.2: Gramian Angular Field output on Oliveoil data

4.5 Encoder

The encoder algorithm solves the lack of pre-trained models for image classification with fewer training images, eliminating the need to appoint a category. Autoencoders use semi-supervised learning algorithms that combine the strength of unsupervised and supervised learning algorithms.

4.6 Multiscale Convolutional Neural Network

The multiscale convolutional neural network extends the dynamism of the hidden layers. Additional convolution layers produce a very coarse output to match the low possibility of components. This thereby accelerates the convergence and increases the stability of the neural network.

4.7 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, more is needed to judge models using accuracy alone thoroughly. 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.

4.8 Experiment

The training and testing sets are provided separately to ensure that the results of several different algorithms of different studies are comparable. Different datasets from the UCR time series were considered to assess the performance of different models. We then compared the RP-CNN with three other benchmark methods. The hyperparameters of the RP-CNN are phase space dimensions. RP (with the classical phase space dimension $m = 3$, and embedding time delay $\tau = 4$) time-series to image encoding on the first sample of different datasets from the considered UCR archive are shown. When the RP images are more significant, they are re-sized to an acceptable size to avoid information loss. The results of compared algorithms are obtained from their respective papers.

4.8.1 Classification Error Rates

Dataset	**RP-CNN**	**Resnet**	**Encoder**	**MCNN**
Wafer	**0**	0.003	0.4	0.087
Olive Oil	**0.11**	0.133	0.6	0.62
Coffee	**0**	**0**	0.021	0.486
Trace	**0**	**0**	0.04	0.646
Face Four	**0**	0.068	0.185	0.732
Two Pattern	0.4935	**0**	**0**	0.597
Adiac	0.28	**0.174**	0.516	0.978
Gunpoint	**0**	0.007	0.064	0.487
Syntheitic Control	0.3433	**0**	0.004	0.702
Swedish Leaf	**0.06**	0.042	0.07	0.882
Average Rank	0.1287	0.0427	0.19	0.6217
Number of Wins	7	5	1	0

4.8.2 Table Interpretation

The table comprises datasets with multistage neural networks. The recurrence plot CNN technique, Resnet, Encoder, and MCNN error rates are documented. The average rank system is used to rate the comparison rate of each. The number of wins shows the number of times a particular algorithm was better than the others.

4.9 Results

The best performer of each dataset is highlighted in bold. RP exploits CNN's high performance on all image classifications. However, Time-series signals are first transformed into images (using RP) and then handled by a multistage deep CNN model.
The table comprises the summary of the performance of all the error results of all the considered algorithms. The better the algorithm, the smaller the average rank. It was noted that RP-CNN had the second-best accurate algorithm with an average rank of 0.1287 and many wins of 7. This was followed closely by Encoder, with an average rank of 0.19 with one win. MCNN had the least accuracy, with an average rank of 0.6217. Resnet had the best accuracy with an average rank of 0.0427 and 5 wins.

4.10 Observations

Recurrence plot transformation techniques are very effective for TSC. Some factors lead to effectiveness. Multistage recurrence plots inherit the advantages of RP, e.g., exposing recurrent patterns and constructing long-term time dependencies. These advantages are highly complementary to CNN. CNN handles the weaknesses of RP, e.g., tendency confusion and long encoding sequences. Therefore, they provide better representations for the time series.

Chapter 5

Conclusion

In multistage modeling using Deep learning, image processing can be two-dimensional or one-dimensional. They can be implemented on a time series classification because time series have an intense time locality that convolutions can extract. Multivariate time series, however, has the same 2D data structures as images after transformation. CNN images are very convenient for processing multivariate time series. CNN can get familiar with an alternate degree of time series features in a classification state. RP can give a decent representation of the m-layered stage space direction, giving the best performance in the experiment. It is noted that the RP-CNN performs well on some datasets, but it can be predicted that deepening the stature of a time series structure may be very good but should be done carefully. Learning hierarchies of concepts and building up multiple layers of abstraction should be fundamental in most deep-learning models. Furthermore, if this is well implemented, very good and time considerate models can be made. Resnet was the best with regard to the datasets chosen. However, when a model is needed for a particular task, several algorithms should be considered, and the algorithm with the best wins and average rank should be considered. In conclusion, high accuracy and high scalability make the perfect algorithm for most deep-learning projects. Many algorithms can be used, and by the ranking and number of wins method, the best one in terms of accuracy and time metric can be chosen. There should be more than one standard algorithm for some problems.

5.1 Future Work and Recommendations

- Echo state Networks can be considered because they can speed up the training process since they are sparsely connected with most of their weights fixed.

- Large datasets are needed to train deep learning architectures well. Therefore, using the proposed pipeline for TSC with very small sample sizes can be another future direction that should be considered.

References

[1] M. C. Mariani , J. Arthur, O. K. Tweneboah (2022) Time Series Classification using Deep Learning

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

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979.

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

[5] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, Y. Liu Deep learning for sensor-based human activity recognition: overview, challenges, and opportunities ACM Comput. Surv., 54 (4) (2021), pp. 1-40

[6] Saint-Etienne, "Classification of Time-Series Images Using Deep Convolutional Neural Networkss," *Saint-Etienne3rd ACM S*, Shaker Heights, Ohio, 1971, pp. 151–158.

[7] Chen, S. Su, H. Yang Convolutional neural network analysis of recurrence plots for anomaly detection Int. J. Bifurc. Chaos, 30 (01) (2020), pp. 201-213

[8] T. Ying, Y. Shi Data mining and big data IEEE Trans. Knowl. Data Eng., 26 (1) (2016), pp. 97-107

Curriculum Vitae

James Arthur was born on April 1988. He graduated from University Of Mines and Technology, Ghana, in 2011. In the fall of 2020, he entered the Computational Science program of the University of Texas at El Paso. While pursuing a master's degree in Computational Science, he worked as a Teaching Assistant. He was a member of African Students Organization. He was a presenter and has a published work by the Hawaii University International Conference.