A Report

On

Multivariate Time Series Analytics

And Its Applications

By

Yash Gupta 2019A7PS1138P

Prepared in Partial Fulfillment of the Course

CS F320 Foundations of Data Science

Birla Institute of Technology and Science (BITS), Pilani

November, 2021

Table of Contents

Introduction

To a beginner in data science, it may seem that 'Multivariate Time Series Analytics' might be a relatively obscure field, existing almost entirely in the domain of research and hardly relevant to what professional data scientists do in practice. After all, don't we deal with images, natural language text, and tabular data of discrete entities most of the time? This, however, cannot be further from the truth. Multivariate time series are everywhere - whether we actively realize it or not. It is not a field restricted to analysis by stock-market experts and economists.

Audio and video data - which have gained much attention in recent years - can be modeled as time series. The pressing need of devising efficient ways to deal with multivariate time series is perhaps even more strongly expedited by the recent advances in the field of IoT. These devices sense information about their environment or user perpetually, and it is upto the data scientists to prevent this data from going to waste. After all, in a data driven world, "data" is the veritable *"oil"* of the 21st century [1].

The objective of this report is to explore the field of multivariate time series (MVTS) analytics. In order to accomplish this goal, a bird's eye view of the prominent applications of MVTS shall be provided. To understand how to actually handle MVTS data, two problems shall be taken up - one involving classification of a time series into a finite set of distinct classes and other being the problem of forecasting future values based on a time series.

First, however, let us briefly see what multivariate time series are and what are the tasks that are commonly associated with them.

1. A Brief Overview of MVTS

The section briefly describes what is meant by the term “multivariate time series”, how it is related to yet different from the more traditional univariate time series, and what are the tasks commonly associated with MVTS. Specifically, we will be exploring the terms *forecasting*, *classification*, *clustering*, *anomaly detection*, and *segmentation* in relation to time series analytics.

1.1 Multivariate Time Series

A *time series* is a sequence of observations on a variable measured at successive points in time or over successive periods of time [2]. The measurements may be taken every day, minute, a fraction of a second, or any other regular time interval. The data points are indexed in the temporal order of their occurrence. Time series analysis is a set of methods to analyze time series data to extract meaningful statistical information [3].

Whether the data is classified as a univariate time series or a multivariate time series depends on the dimensions of each data point in the time series. Specifically, in a univariate time series, each datapoint is a scalar, as opposed to a vector, or more generally, a higher order tensor.

A multivariate time series, by contrast, contains as a data point in time a higher order tensor. The simplest case would be a time series of vectors.

If $v_t^T = \left[x_{1t}\, x_{2t} \dots x_{nt}\right]$ be a row vector with index t in time, then each attribute of the vector $x_{1t}, x_{2t}, \dots, x_{nt}$ may be considered a variable in its own right, hence the name *multivariate*.

The question arises - can't we just split a multivariate time series $\{v_{t_0}, v_{t_1}, \ldots\}$ into univariate time series of the components - $\{x_{1t_0}, x_{1t_1}, \ldots\}, \ldots, \{x_{nt_0}, x_{nt_1}, \ldots\}$ - and analyze them separately? While it may be possible in some cases, more often than not, the individual attributes are correlated to each other, necessitating that we analyze them together, taking into account not only the past values of that attribute but also those of the attributes closely related to it. For instance, it only makes sense to analyze the pressure and temperature of a climatic zone together, for they are often inversely related.

Now let us briefly go through some tasks prevalent in MVTS analytics.

1.2 Forecasting

To put it simply, a *forecast* is simply a prediction of what will happen in the future [2]. Scientific predictions are made based on historical time stamped data, using models built through historical analysis. At the time of the work, the future outcome is completely unavailable and can only be estimated through careful analysis and evidence-based priors [4].

The simplest mode of prediction - albeit not useful - would be to make a random guess in the interval of allowed values. Time series data, however, allows us to do better. The prediction for the next time interval F_{t+1} can be expressed as some function of the actual prior observations $Y_t, Y_{t-1}, \ldots Y_1$ and the previous forecasted values $F_t, F_{t-1}, \ldots, F_1$. In case of a multivariate time series, all these values are arrays as opposed to numbers.

One of the commonly used techniques for multivariate time series forecasting is the Vector AutoRegression (VAR) model. It will be discussed in Section 4, where it will be used to forecast future air quality indices based on the collected `.csv` MVTS data.

Another method of forecasting is based on neural networks. A Recurrent Neural Network (RNN) is well-suited to time series data. In an RNN, the links - in addition to connecting forward - may connect nodes within the same layer or from the previous layers [5].

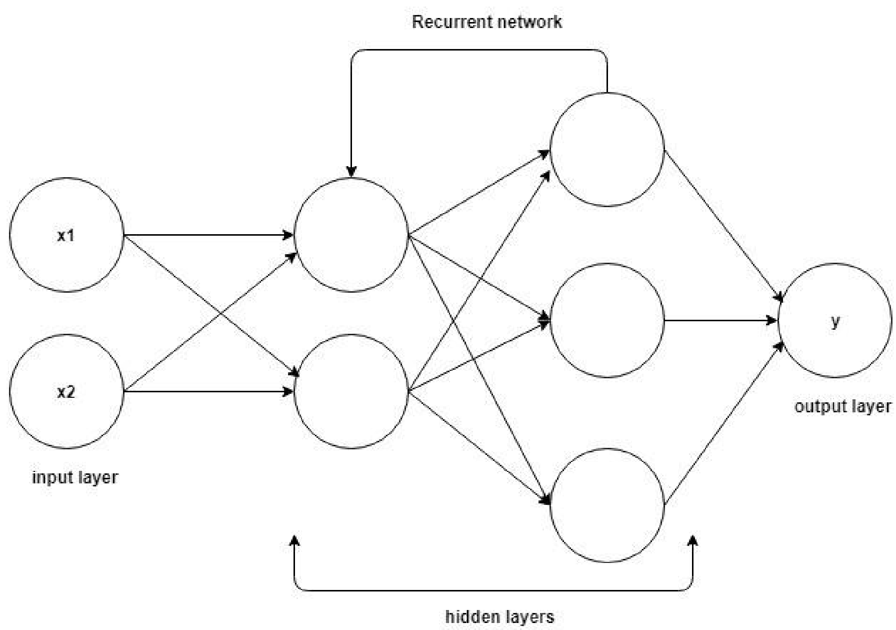

Figure 1.1 The depiction of a basic RNN.
Image Credit: Debarko De, Medium.

An RNN must satisfy the following three properties to be useful [6]:

- The model must be able to store information for an arbitrary duration.
- It should be resistant to random fluctuations or noise.
- The parameters should be trainable in a reasonable amount of time.

By maintaining a "*memory*" of prior predictions, an RNN effectively uses both the latest input and the prior time series data to make the forecast.

1.3 Classification

Classification is the task of assigning an object to one of the several predefined categories [5]. The object, in the case of MVTS analytics, is a multivariate time series itself.

Thus the task becomes: given a set of multivariate time series and corresponding labels, $\{(TS_1, y_1), (TS_2, y_2), \ldots (TS_m, y_m)\}$, develop a model M which when input a time series TS, produces a label $\widehat{y}$ for the same. Note that here each $TS = \{v_{t_1}, v_{t_2}, \ldots, v_{t_n}\}$ is a time series rather than a single attribute vector v.

The need for such a model arises commonly when faced with the task of classifying a complex time series entity as a whole - as in audio or video genre / type recognition and human activity recognition from sensor data or video data. It can also be useful in stock market prediction, where the time series for a stock can be used to infer whether the stock is risky to invest in.

Section 3 describes a few models and techniques to accomplish the task using the problem of human activity recognition using information gathered by accelerometers and gyroscopic sensors.

1.4 Clustering

Cluster analysis divides data into groups, or clusters, that are meaningful, useful, or both. It groups the objects based only on the information found in the data that describes the objects and their relationships. The goal is to have the objects within a group similar to each other and different from other objects in the group [5].

The objects in case MVTS analytics are - as was the case with classification - the multivariate time series themselves. Now how do we measure the degree of similarity or difference between the time series.

The commonly used Euclidean distance fails in case of time series data, for two time series can give drastically different values for Euclidean distance when phase-shifted - for instance, $sin(\omega t + \frac{\pi}{2}) = cos(\omega t)$, thus the Euclidean distance between $sin(\omega t)$ and $cos(\omega t)$ can be reduced to zero with a $\frac{\pi}{2}$ phase shift.

To resolve this problem, dynamic time warping (DTW) is used with the objective of comparing arrays with different length by building one-to-many and many-to-one matches so that the total distance is minimised.

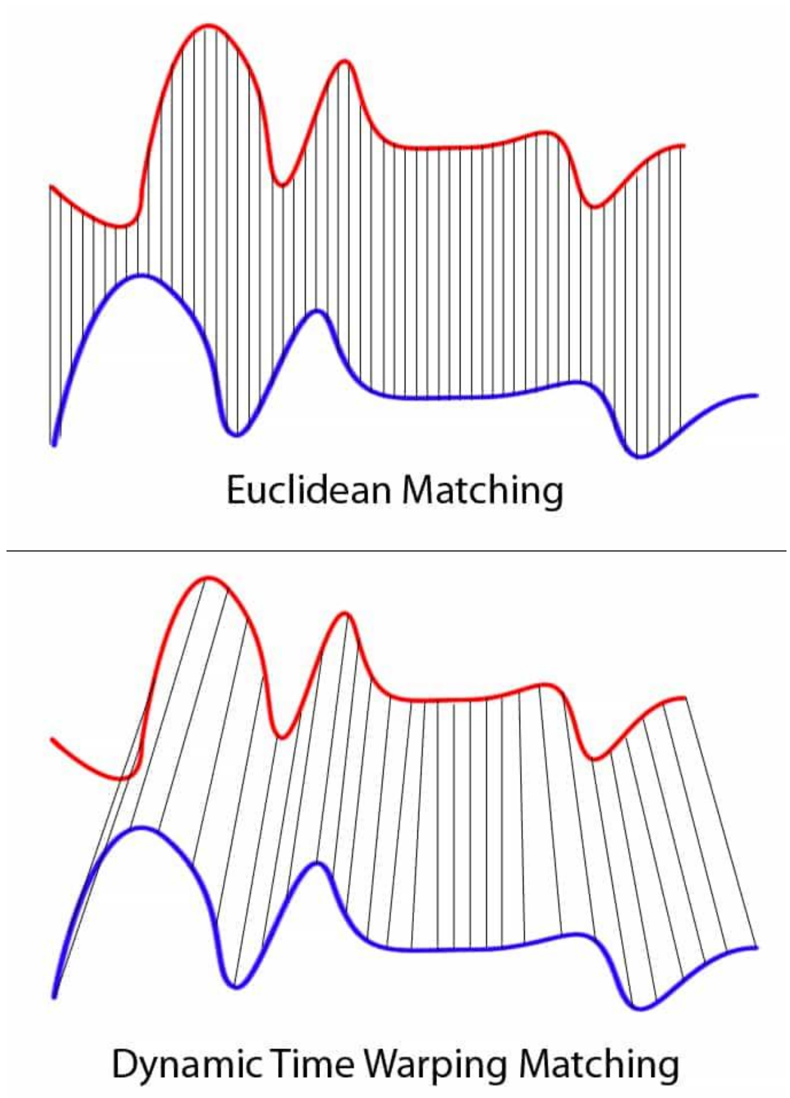

Figure 1.2 Euclidean vs DTW Matching.
Image Credit: Databricks.

Using DTW to give a measure of *"distance",* clustering algorithms such as the k-medoids may be used - k-means does not minimize DTW [7] - to form clusters of related time series.

Clustering finds applications in grouping similar sounds - in terms of tone and exaggeration - together or in clustering similar stocks for a more structured further analysis by experts.

1.5 Anomaly Detection

The goal of anomaly detection - or deviation detection - is to identify objects that are different from most other objects, sometimes called outliers or anomalies [5]. The object, in case of MVTS time series analytics, is a multivariate time series considered as a whole. The goal can be achieved in multiple ways - classification, clustering, or statistical outlier detection.

By using labeled data, we may train a classifier, which assigns a label - *anomaly* or *not an anomaly* - to an object, thereby reducing the task to that of classification. Thus, we may use methods of Section 3 for anomaly detection.

Using clustering based techniques, small clusters that are different from other clusters may be flagged anomalies. Also, if the elimination of an object from a cluster results in substantial improvement in the cluster similarity, it may be flagged as an anomaly [5].

Finally, a datapoint that has a low probability of occurrence with respect to the probability distribution model of the data may be marked an anomaly. We used $z = \frac{x - \mu}{\sigma}$ in case of simple numeric data to estimate the probability of occurrence of the datapoint x. In the case of MVTS, DTW may be used to measure the distance from the center of the data.

Anomaly detection may find use in detection of unusual sounds in audio data or unusual seismographic or weather patterns in meteorology.

1.6 Segmentation

The process of time series analysis can be greatly simplified by first segmenting, or dividing, the time series into homogenous regions. Then, the entire time series can be captured by presenting only k short representative patterns to a human annotator in order to produce labels for the entire dataset, where k is the number of segmented regions, or *regimes* [8].

For instance, the macroeconomy of a country can be understood in terms of discrete homogenous periods, showing behaviour such as a "boom" or a "slump". A human reader would find it much easier to comprehend such a representation than a long time series of numbers, or vectors. This process of simplification is called segmentation, and is generally achieved with a matrix profile, using tools such as `STUMPY`.

2. Applications of MVTS

The section enumerates some of the prominent applications of multivariate time series, with the goal of highlighting the need of MVTS analytics in the arsenal of every data scientist.

2.1 Stock Prices Forecasting

The knowledge of the future, arguably, can not make as much impact in any other field as it does in the domain of stock market and share analysis everyday. The information about tomorrow's - or even an hour later's - stock prices can make a difference of millions for many investors and brokers.

MVTS analytics can be put to use in stock price forecasting. Given the data of the form shown in Table 2.1 below, it is possible to predict the future values of the corresponding attributes using a forecasting system such as an LSTM [9].

Table 2.1 Stock Market Data as available through Google Finance API.

Date	Open	High	Low	Close	Volume
06/30/2017	943.99	945.00	929.61	929.68	2287662
...	...	...	...	...	...

Open:	Starting price at which the stock was traded on a particular day
Close:	Final price at which the stock was traded on a particular day
High:	Maximum price of the share for the day
Low:	Minimum price of the share for the day
Volume:	Total number of shares processed on that day

While it is possible to analyze the time series for attributes like 'Open' separately, it only makes sense that the opening price of the next day might be correlated to the closing price of the last. Hence, MVTS forecasting is the approach to go for.

2.2 Renewable Sources of Energy

The measurements obtained through various weather instruments - such as, barometer for pressure, hygrometer for humidity, and anemometer for wind speed - may used to form a multivariate series, where each data point in time is a tuple of weather parameters - $(temperature,\ dew\,point,\ pressure,\ humidity,\ wind\,speed)$.

The parameters like wind speed and pressure are relevant to quantify renewable energy production in a wind turbine. These may be estimated by performing a univariate time series forecast on wind speed data using means like AR, MA, ARMA, and ARIMA. However, it may be noted that there exist relationships between various weather parameters - for instance, pressure and temperature are inversely related. Therefore, a MVTS forecasting using Vector AutoRegression (VAR) - or other multivariate counterparts like Vector AutoRegression Moving Average (VARMA) - is a more appropriate method to gain an estimate of wind energy production in the coming days.

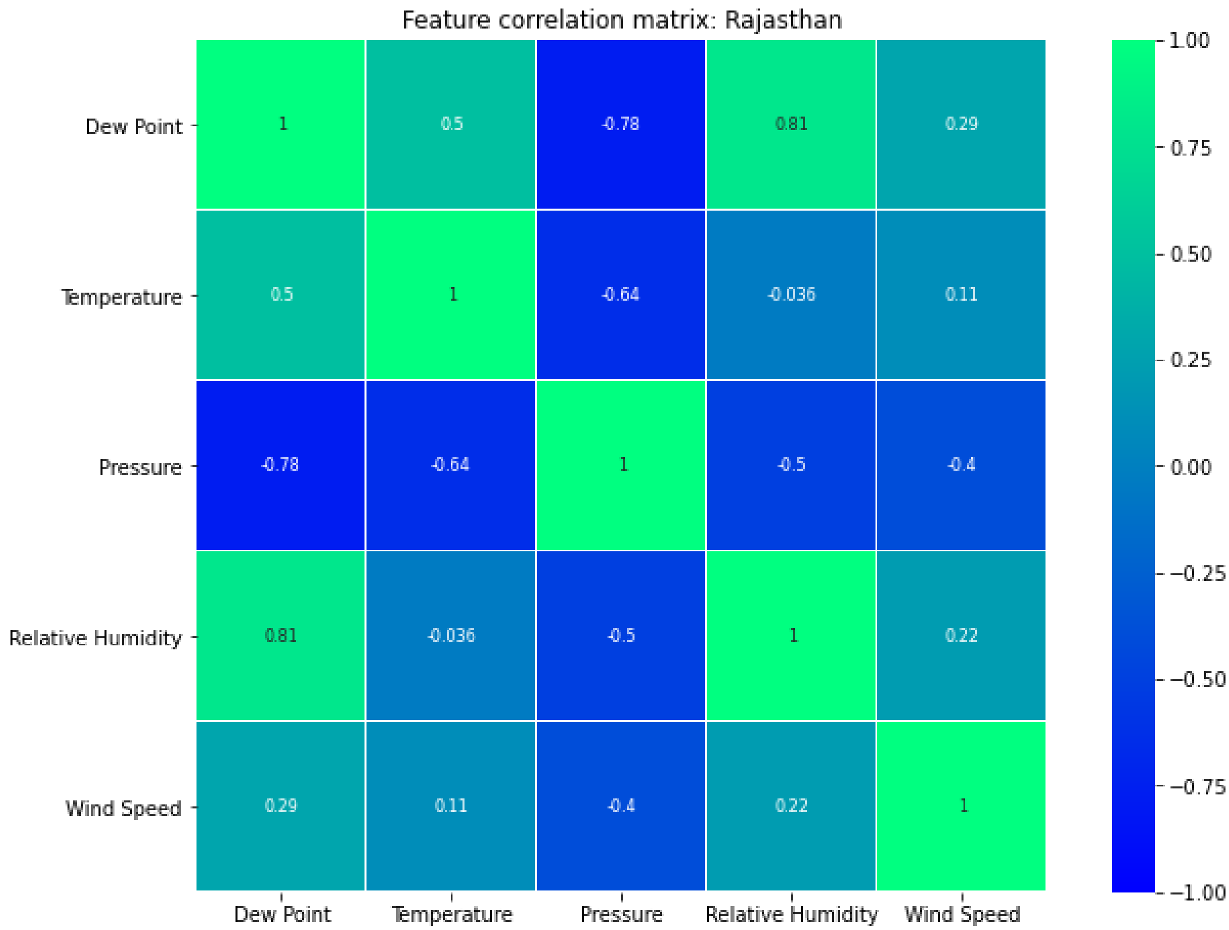

Figure 2.1 Correlation between Weather Parameters (e.g. -0.64 for pressure and temperature)
Image Source: Math F432 Assignment - 2

Such information about renewable energy production - wind energy, in this case, but could be solar energy or tidal energy in others - could prove invaluable to energy distribution planners who have to plan backup measures to meet community power requirements, in case the clean power supply falls short of the demand.

2.3 Character Trajectory Analysis

A WACOM tablet can be used to record the time series of tuples (x, y, F) - x being the x-coordinate, y being the y-coordinate, and F being the pen tip force - for a character as it is drawn by a volunteer [10]. Such data could then be subjected to multivariate time series clustering or classification to identify similar characters or to recognize a character from its trajectory.

Figure 2.2: Trajectory for ‘a’ - a MVTS of (x, y, F)
Image Credit: Wikimedia Commons

By recognizing a character from its trajectory as it is being drawn, the tablet can offer auto-complete suggestions for the character to the user, thereby improving the user experience. A cluster analysis can help human graphologists in studying the evolution of cursive handwriting.

2.4 Electrical Biosignals

Electrocardiograms (ECG), Electroencephalograms (EEG) and Magnetoencephalography (MEG) are all multivariate time series of several electric signals of several bands that measure - directly or indirectly - actual or relative changes in voltage throughout the body. ECGs are typically used to detect and measure the electrical activity of the heart. EEGs are used to measure brain activity (brain waves), and are typically used in the diagnosis of epilepsy and seizures. MEGs record the magnitude of the magnetic field produced by the brain [11].

Such signals can be used to train classifiers for performing tasks as diverse as recognizing the direction of hand movement, predicting the termination or non-termination of atrial fibrillation, and identifying parts of the brain associated with the movement of the tongue [11].

2.5 Quality Control in Solutions

EthanolConcentration is a dataset of raw spectra taken of aqueous solutions of ethanol $C_2H_5OH\,(aq)$ in 44 distinct, real whisky bottles, with the alcohol concentrations 35%, 38%, 40%, and 45%, which will serve as class labels. Producers are required to ensure that their products contain alcohol concentrations tightly bound to what is reported on labelling. The classification problem is to determine the alcohol concentration of a sample contained within an arbitrary bottle [11].

The raw spectra data in this case comprises a multivariate time series using wavelengths in the range $226\,nm$ to $1101.5\,nm$ with a StellarNet BLACKComet-SR spectrometer. This MVTS data is used to train a classifier, which when given a new spectra data, predicts the concentration class of the ethanol solution. This enables the producers to ensure that their alcohol is satisfying the quality control restraints of matching the reported concentrations.

2.6 Recognition of Astronomical Bodies

Large Synoptic Telescope (LSST) views an object with different LSST passband integers - u, g, r, i, z, Y = 0, 1, 2, 3, 4, 5$u,\ g,\ r,\ i,\ z,\ Y\ =\ 0,\ 1,\ 2,\ 3,\ 4,\ 5$ - and measures the brightness, or flux, of the object as a time series [12]. This multivariate time series of brightness under different passbands, or astronomical filters, can be used to train a classifier to detect the various classes of astronomical objects.

Therefore, a task that was once only possible through manual observations and inspection by human astronomers can now be automated through a MVTS classifier.

2.7 Audio Data

While audio data can be analyzed as a univariate time series in time domain, it is often a more powerful technique to represent it as a multivariate time series in a spectrogram format. This format exposes the spectral decomposition of the data and expresses the change in spectral power over time [11].

Examples of MVTS analytics on audio would be genre classification for songs and identification of the bird responsible for a particular sound from an audio file.

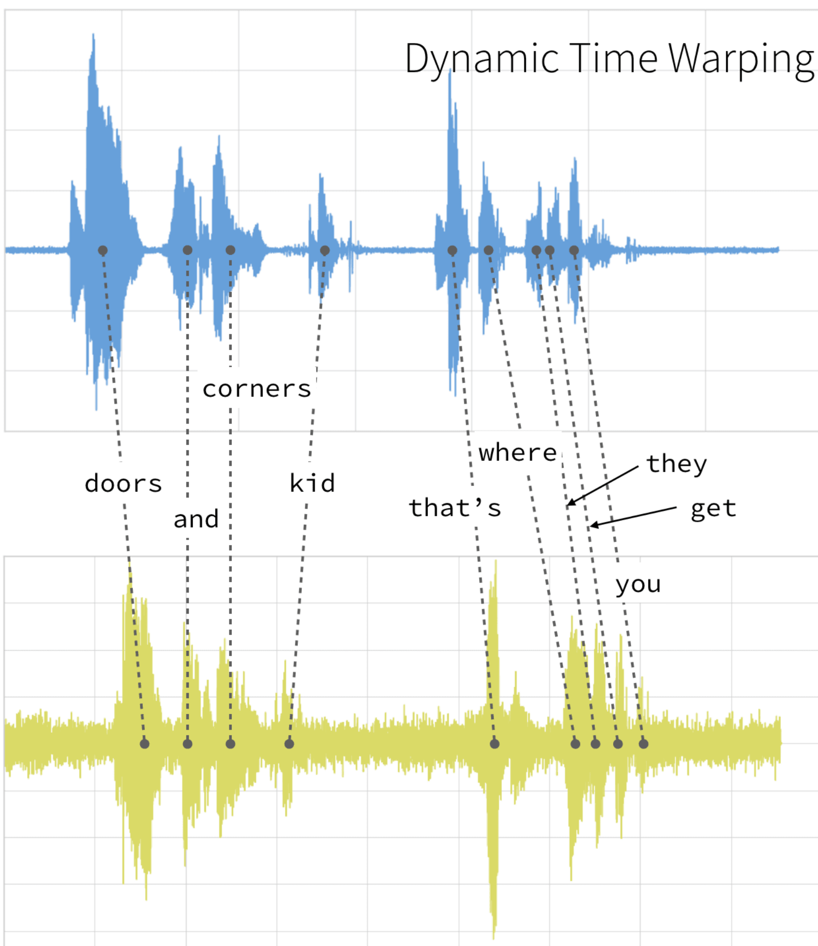

Figure 2.3 Measuring Similarity between audio files.
Note this representation is a univariate series - can be converted to a multivariate spectrogram.
Image Credit: DataBricks

Using algorithms like DTW, we can also cluster similar audio files together, as illustrated in Figure 2.3 (figure shows a univariate time series representation, but it can be converted to a more powerful multivariate time series format).

Heartbeat sounds could be analyzed to detect anomalies in patients, signifying that the patient needs further diagnosis.

2.8 Sign Language

The datasets, such as AUSLAN (Australian Sign Language signs data set), can make lives for countless people with accessibility problems in hearing or speaking much easier. A Nintendo power glove can be used to collect tuples at regular intervals time containing information about the hands of the wearer - moving left or right, moving up or down, moving backwards or forwards, and palm pointing up or down. This information about the signs - such as that depicted in Figure 2.4 below - can be expressed as a multivariate time series.

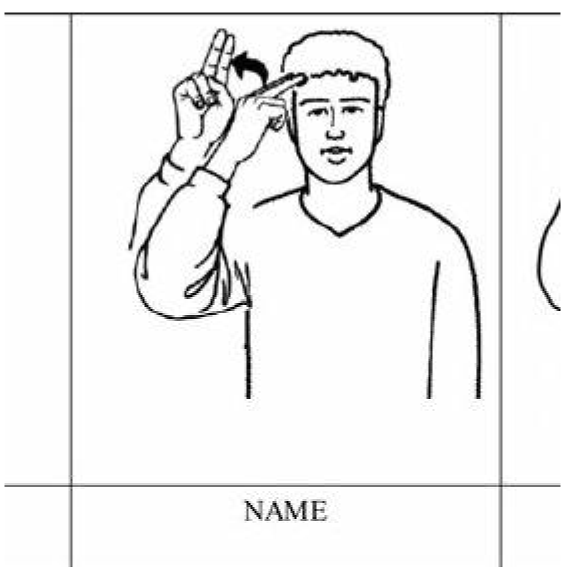

Figure 2.4 Sequence of movement of hand for “Name” in AUSLAN.
Image Credit: Phonological variation and change in Australian and New Zealand Sign Languages: The location variable, ResearchGate.

Such information can be used to train a classification model which can then translate a sequence of movements of hand and fingers to a word or a phrase, which may be understood by someone new or alien to the language.

2.9 Macroeconomics

To model complex economic phenomena such as inflation and wage growth rate, a univariate time series is hardly adequate. Yash. P. Mehra's 1994 article uses a multivariate time series for analysis - though he uses traditional statistical methods and domain expertise.

The multivariate time series consists of the features [13]:
- *rgnp* : Real Gross National Product (GNP)
- *pgnp* : Potential GNP.
- *ulc* : Unit labor cost.
- *gdfco* : Fixed weight deflator for personal consumption expenditure excluding food and energy.
- *gdf* : Fixed weight GNP deflator.
- *gdfim* : Fixed weight import deflator.
- *gdfcf* : Fixed weight deflator for food in personal consumption expenditure.
- *gdfce* : Fixed weight deflator for energy in personal consumption expenditure.

The time series can be used to forecast the feature variables for the future, using methods like VAR and LSTMs. Such information can prove crucial to a country's finance ministry who can plan the fiscal policy for the upcoming year with a view to avoid situations like the Great Depression and Zimbabwe's hyperinflation.

2.10 Robotics

MVTS can help in model the behavior of robots by monitoring different physical parameters - torque, power, force, momentum, and position - in time. To understand it better let us consider the following example.

The quintuples (F_x, F_y, F_z, τ_x, τ_y, τ_z) were recorded for a robot after its failure. Here F_x and τ_x represent the force and the torque measured along the x-axis, and so on. The goal was to identify the type of failure using MVTS from among the following labels:

- LP1: failed to approach grasp position
- LP2: failed to transfer a part
- LP3: failed to reach correct transfer position
- LP4: failed to approach ungrasp position
- LP5: failed to move the part [14]

A classifier can be trained for MVTS data. Hence, the correct classification of failures in the future can provide valuable insights to the engineers on how to improve their robotic systems, without manually trying to find the type of failure.

2.11 Malware Detection

A robust strategy for detection of and response to malicious code invading a computer system is of the paramount importance in an age where almost every household has a miniature computer in the form of a smartphone, every banking transaction is conveniently carried out online, and students share their personal information for examination applications. Though various malware detectors have been developed in the past using flag features or a univariate time series of API calls, MVTS can also be used effectively for malware detection.

Ki-Hyeon Kim et. al (2015) [15] proposed a malware detection model for Android smartphones using a multivariate time series with the features shown in Figure 2.5 to monitor system usage for unusual trends.

TABLE I. Linux kernel-based proposed feature

Category	Feature
Memory	Total_CPU, User_CPU
CPU	Usage_memory
Network	Rxbytes, Txbytes

Figure 2.5 Features for malware detection for Linux-Kernel based Android smartphones.
Image Credit: See *Reference 15.*

Using the information, an anomaly detection system was proposed, which was able to achieve an accuracy of 0.989 for the task.

2.12 Space Weather Data

Certain astronomical phenomena, especially those occurring in the solar system, have a profound impact on life on earth and the human satellite system. For instance, solar flares pose a hazard to sensitive space equipment and astronauts. In addition, they can lead to surges, tripping, and even melting of earthbound transformers [16].

A MVTS of solar flare predictive parameters, such as those measuring magnetic flux from solar corona available from Solar Dynamics Observatory (SDO), can be used to predict the occurrence of solar flares by detecting patterns similar to those in past data [16]. Here clustering and classification for MVTS data can be of immense help. Further, the values of certain parameters, like those measuring solar radiations, can be forecasted to get an idea of intensity of the next major solar flare.

3. Problem #1: Human Activity Recognition

3.1 The Problem

With the advent of specialized IoT sensors in smartphones it has become possible for individuals to monitor their activity rate or body health parameters. The sensor information can be used to form a time series of observations about the motion of the body - linear and angular kinematics - which can then be used for a multitude of tasks, one of which is Activity Recognition.

Activity Recognition refers to the task of identifying the actions or objectives of one or more agents from a series of observations. It finds applications in providing rehabilitation to people with brain traumas and security monitoring applications [17].

The goal of the problem is to use the acceleration and gyroscopic information in the form of an MVTS to develop a classification model to identify the underlying overall activity performed for a given MVTS of similar form.

3.2 The Dataset

The "Human Activity Recognition Using Smartphones Data Set", available at this link, has been used for the task. It contains linear and rotational motion information about 30 volunteers - between ages 19 and 48 - as each of them performed 6 different activities wearing a Samsung Galaxy S II smartphone on their wrist.

The 6 activities will serve as the classes for the classification task - WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, and LAYING.

3-axial linear accelerations and 3-axial angular velocities measured by the sensors over time were preprocessed by the researchers by applying noise filters and then sampled with sliding time windows of 2.56 each with a 50% overlap, thereby giving 128 reading for each window. They then partitioned the volunteers into training and testing sections in the ratio 7:3.

The data ultimately used is captured by the directory structure given in Figure 3.1 below.

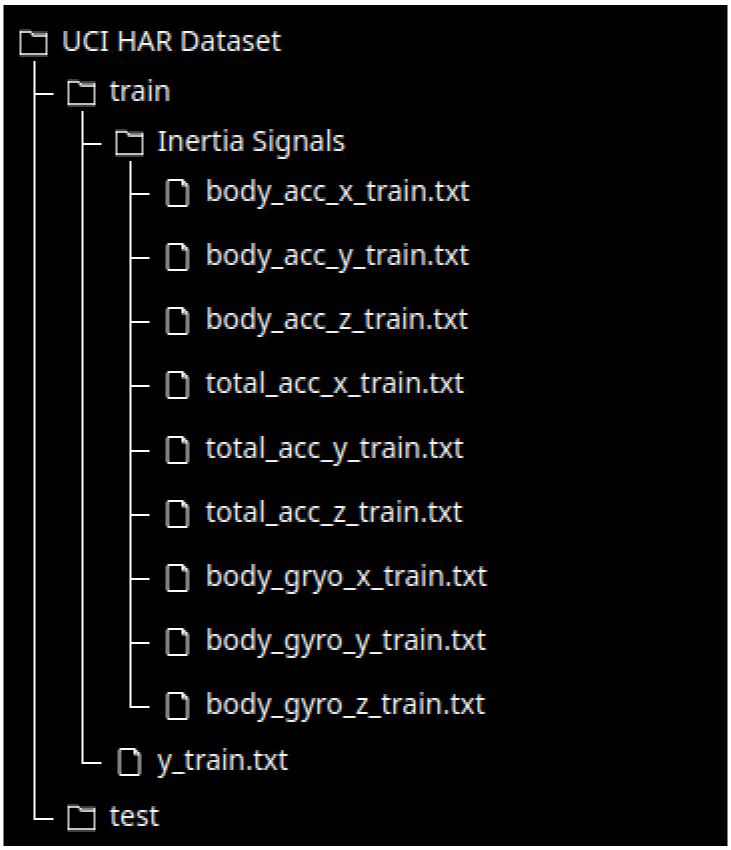

Figure 3.1 Directory Structure for the dataset. ‘test’ has same structure as ‘train’
Image Created using ASCII Tree Generator.

There are 7352 instances in the training set, each with a time series of length 128 with 9 features. The 9 features are the total acceleration in x-, y-, and z-directions; the estimated body acceleration in x-, y-, and z-directions; and the body gyro in x-, y-, and z- directions. Similarly, there are 2947 instances in the training set.

3.3 Data Loading and Preprocessing

Please refer to the notebook MVTSClassification.ipynb for the following sections. The data was uploaded to Google Drive and then loaded with `pandas` in Google colab, reading each of the files and stacking them as feature columns. The form shown in Table 3.1 was used for the data arrays.

Table 3.1 Shapes of the training and testing data arrays.

Array	Shape
`Xtrain`	`(7352, 128, 9)`
`Ytrain`	`(7352, 1)`
`Xtest`	`(2947, 128, 9)`
`Ytest`	`(2947, 1)`

The input arrays - `Xtrain` and `Xtest` - were hence copied and transformed into nested dataframes, as `sktime` routines do not accept `numpy` arrays directly as inputs. Also the label data arrays - `Ytrain` and `Ytest` - were converted to one-dimensional tensors from two-dimensional ones with class label offset changed from 1 to 0.

3.4 MiniROCKET + RidgeClassifier

The original plan for the classification model was to use a classifier based on DTW, an ensemble like the BossEnsemble, or the MrSEQL algorithm. However, such algorithms took 3 hours for training on both standard and accelerated runtimes and still did not finish executing, at which Google Colab environment auto-discarded the training process. Therefore, a pressing need for a faster, yet reasonably accurate, model was felt.

ROCKET, or the RandOm Convolutional KErnel Transform, came as the answer. It is a strategy for feature extraction, which transforms the MVTS to a set of features with information about the class membership of the series. It uses 10,000 kernels for *convolving* with each series, and applies global max pooling and *ppv* - or proportion of positive values - to produce a set of features [11].

Here, $ppv\,(X * W - b) = \frac{1}{n}\sum(X * W - b > 0)$, where X is convolved with weights W and the bias b is subtracted.

However, before applying ROCKET, further research revealed an even faster (~ 75 times faster) version for transformation - MiniROCKET, or Minimally Random Convolutional KErnel Transformation. It makes the following changes to the kernel lengths, weights, bias, dilation, and padding to achieve faster results.

Table 1: Summary of changes from ROCKET to MINIROCKET.

	ROCKET	MINIROCKET
length	$\{7, 9, 11\}$	9
weights	$\mathcal{N}(0, 1)$	$\{-1, 2\}$
bias	$\mathcal{U}(-1, 1)$	from convolution output
dilation	random	fixed (rel. to input length)
padding	random	fixed

Image 3.2 Differences between ROCKET and MiniROCKET kernels.
Image Credit: *See Reference 18.*

The transformed values of inputs - now with 9996 features per instance - were passed for classification with a linear classifier - the Ridge Classifier. A multiclass ridge classifier trains a binary ridge classifier for each class, converting the target values into $\{-1, 1\}$, and then treating the problem as a regression problem. It uses the features to train a Ridge Regression model [19].

A Ridge Regression model is a better choice than the vanilla linear regression for features with significant mutual correlations. It estimates the model coefficient vector $\beta \in R^{nX1}$ for the input feature matrix $X \in R^{mXn}$ using training labels $y \in R^{mX1}$ as $\hat{\beta} = (X^T X + kI_n)X^T y$, where I_n is an identity matrix and $k > 0$ is a small number.

The model $NestedDataframe\ MVTS \rightarrow MiniROCKET \rightarrow RidgeClassifier$ was developed using `sktime` and `sklearn` routines. The `RidgeClassifierCV`, in particular, was used, which uses the leave-one-out cross validation to estimate the best hyperparameters.

3.5 LSTM

A Long Short Term Memory RNN model was also developed for the same task - it can help put the performance of the MiniROCKET + Ridge Classifier model in context. The neural network contains a single LSTM layer with 100 units, followed by a fully-connected dense layer of 100 units with ReLU activation, and finally a Softmax activated layer with 6 neurons - one for each class. The model is summarized by Figure 3.3 below.

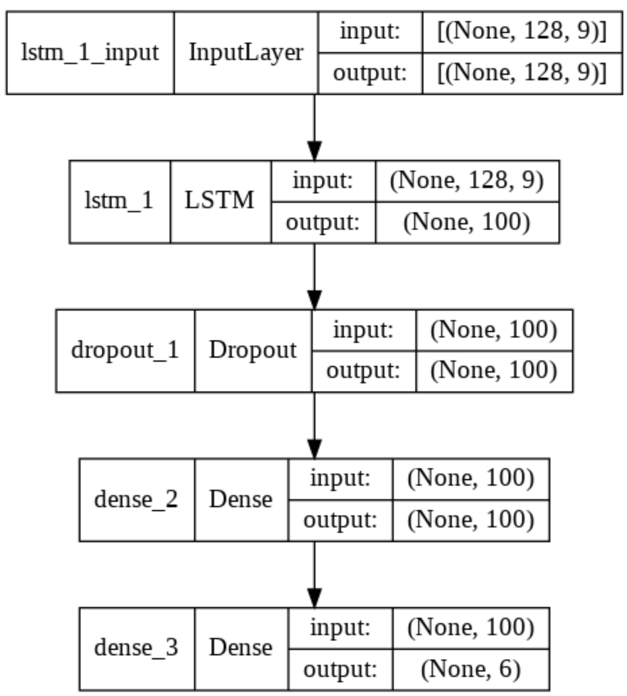

Figure 3.3 LSTM model outline.
Image Source: *See MVTSClassification.ipynb.*

To minimize the effects of overfitting, a dropout layer with drop rate 0.2 was sandwiched between the LSTM and ReLU activated fully connected layers. *[Dropout rates of 0.3, 0.4, and 0.5 were also tried, but they did not have any significant effect on performance as measured by accuracy.]*

The model was compiled with Adam, or adaptive moment estimation, optimizer for computational efficiency in training. Since the task is a multi-class classification, the loss function used was categorical cross-entropy. The model was trained for 20 epochs.

3.5 Results

The model in Section 3.4 - the MiniROCKET + RidgeClassifier - performed surprisingly well, achieving an accuracy of 97.32% on the test dataset under less than 5 minutes of training on Google Colab standard environment, thereby validating claims of ROCKET (and hence MiniROCKET) achieving state-of-the-art performance "remarkably fast" [11].

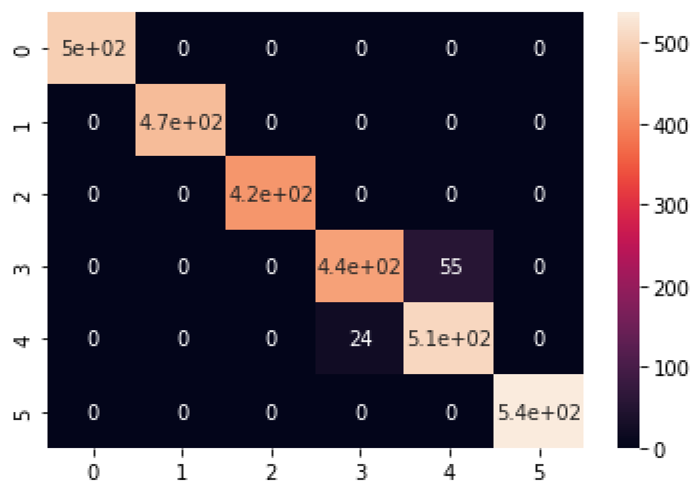

Figure 3.4 Confusion Matrix for MiniROCKET + RidgeClassifierCV on test data.
Image Source: *See MVTSCLassification.ipynb.*

LSTM, on the other hand, achieved a test accuracy of 90.33% after 20 epochs, taking around 10 - 15 minutes for training in the same environment. After about 12 epochs the performance had more-or-less saturated, as measured by accuracy. The slight overfit, minimized by dropout, could be further diminished using regularization.

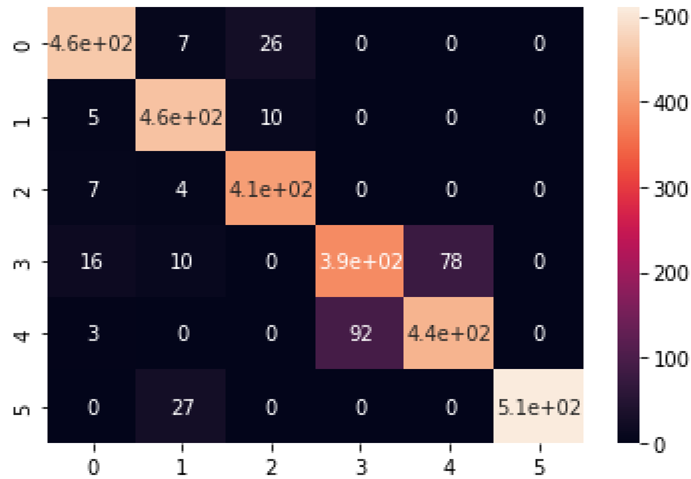

Figure 3.5 Confusion matrix for LSTM model for test data.
Image Source: *See MVTSClassification.ipynb.*

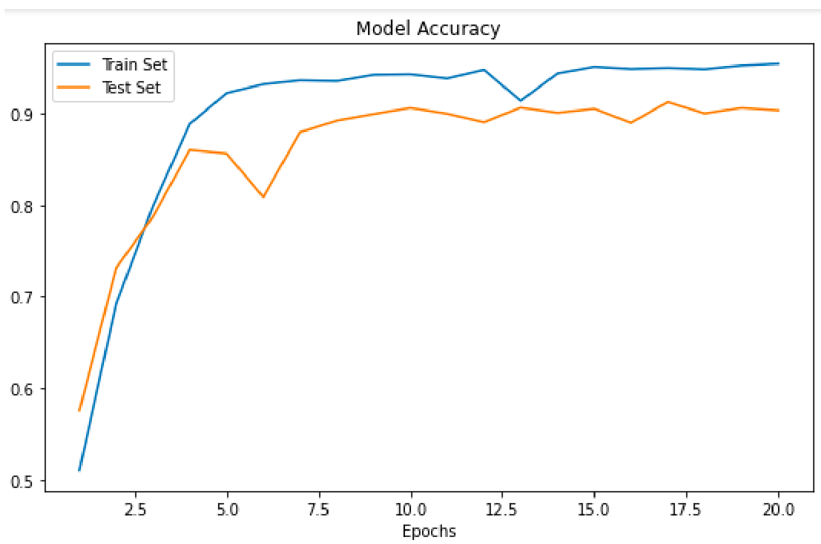

Figure 3.6 Accuracy plot for LSTM. Slight evidence for overfit - could be rectified with regularization.
Image Source: *See MVTSClassification.ipynb.*

In conclusion, MiniROCKET shows great promise in time series classification. It can be run with a multitude of other classifiers, besides Ridge - such as Lasso - in a much smaller time compared to other time series classifiers.

4. Problem #2: Air Quality Forecasting

4.1 The Problem

The current state of affairs of the planet's environment is tragic, to say the least; it presents, nonetheless, a thoroughly challenged field of research and application to data scientists to develop useful models for medical applications (of the likes of air filter regulation), environmental policy making (through analysis of climatic change data), clean energy harvesting (see Section 2.2), and the like.

The present problem is to *forecast*, or predict, the air quality, as measured by concentration of air pollutants, for a city based on a multivariate time series of observations of concentrations and basic environmental parameters in the past. I personally feel that such a model can be of great use to city planners in metropolises like New Delhi, which are perpetually close to alarming levels of pollution.

4.2 The Dataset

The "Air Quality Data Set", available at this link, has been used for this problem. The data summarizes in a `.csv` file the concentrations of the pollutants - carbon monoxide (CO), non-metanic hydrocarbons, oxides of nitrogen, and benzene - along with the temperature and humidity - absolute and relative - conditions.

The 9358 instances were recorded using an array of metal oxide - tin oxide, titania, tungsten oxide, indium oxide - chemical sensors for an Italian city for the period March 2004 to February 2005. Thus, it is a single multivariate time series of 9358 tuples of concentrations and environmental conditions (a total of 13 features) *[Please refer to VectorAutoregression.ipynb for further details]*.

4.2 Data Loading and Preprocessing

The data file was read using `pandas.read_csv()` to generate a data frame containing the MVTS. The empty rows and empty columns were eliminated as the first step of data cleaning. The researchers used the value -200 to represent the null values. The values were approximated by the previous value for that feature to obtain a form suitable for Vector AutoRegression operation. The specific date and time values were replaced by indices for the time series tuples.

4.3 Analysis and Forecasting

First step was to decide whether to use a vector autoregression or to use univariate time series forecasting methods - such as AR, MA, ARMA, and ARIMA - for the features separately. A correlation matrix (show in Figure 4.1) was plotted for the features, and a significant degree of correlation was observed for certain pairs of features - for example $\rho(PT08.S5(O_3), PT08.S2(NMHC)) = 0.88$. Such relationships necessitate a MVTS analysis of the data.

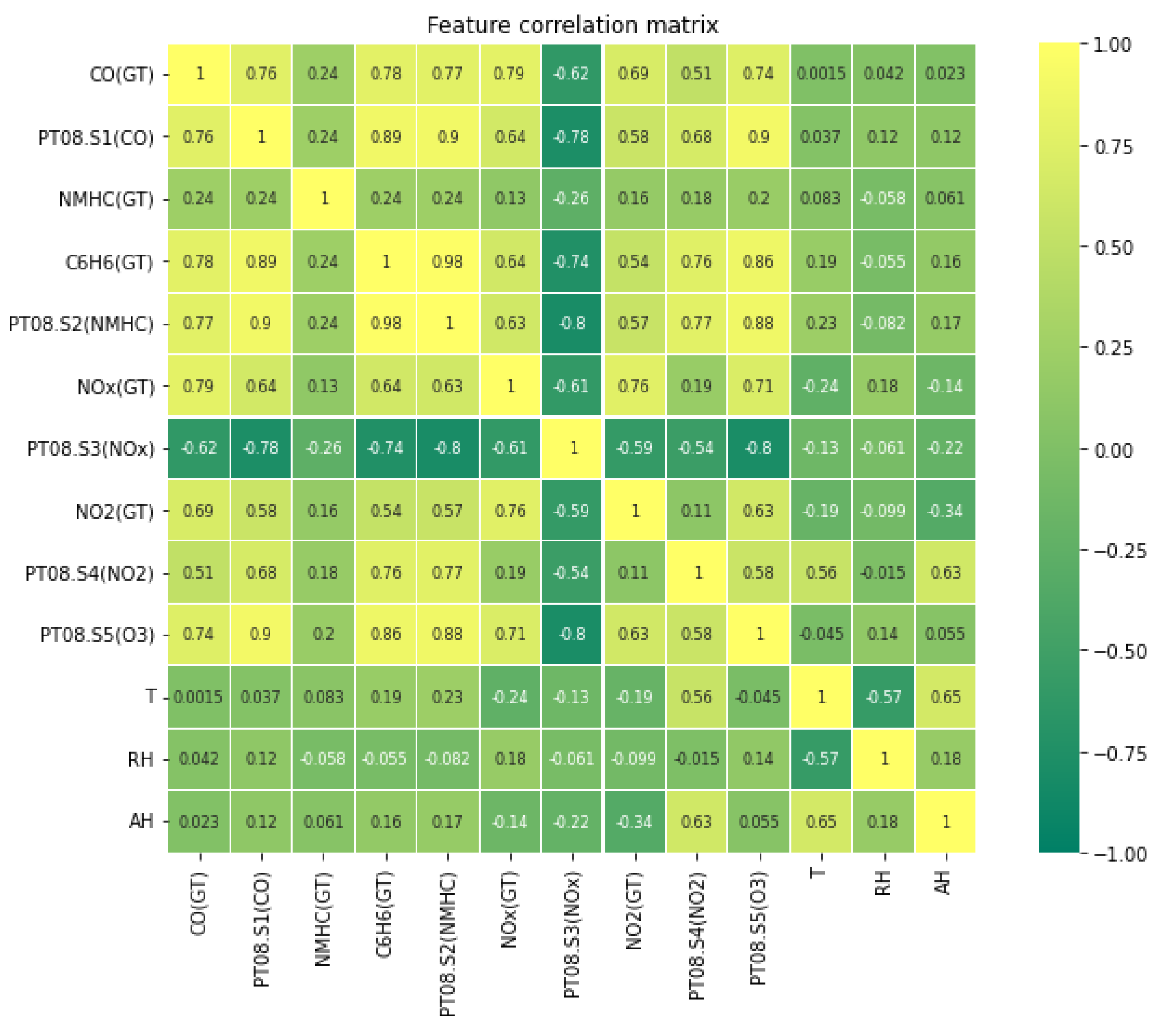

Figure 4.1 Correlation Matrix for AirQualityUCI data features.
Image Source: *See VectorAutoregression.ipynb.*

We plan to use Vector AutoRegression (VAR) if the time series satisfies certain conditions, stated further below. VAR, of lag order p, can be summarized as:

$v_t = \theta_0 + \theta_1 v_{t-1} + ... + \theta_p v_{t-p} + \epsilon_t$, where v_t is the predicted vector at time t, θ_0 the intercept, $\{\theta_1, ..., \theta_p\}$ the coefficient matrices, and ϵ_t the white noise. ϵ_t should be a continuous random variable (vector) with $E(\epsilon_t) = 0$ and $E(\epsilon_{t_1}, \epsilon_{t_2}) = \sigma_{12}$, the standard deviation of the series [20].

This can be represented in terms of lag operators $Lv_k = v_{k-1}$ as follows:

$v_t = \theta_0 + \theta_1 Lv_t + ... + \theta_p L^p v_t + \epsilon_t$, which reduces to

$(I - \theta_1 L - ... - \theta_p L^p)v_t = \theta_0 + \epsilon_t \Rightarrow \Phi(L)v_t = \theta_0 + \epsilon_t$.

Therefore, $v_t = \Phi(L)^{-1}(\theta_0 + \epsilon_t)$.

For the inverse to exist, $det(\Phi(L)) \neq 0$, that is, the absolute values of the eigenvalues must be all less than 1.

This condition will be used to test whether VAR is applicable to the data or not using the Johansen cointegration test. This test allows us to determine whether the series has an underlying test - VAR not applicable directly - or is stationary. For the present case, the series does not show any trend and we apply VAR.

The data is partitioned for training and testing in the ratio 9:1. We search for an optimum lag order p (using Akaike Information Criterion) for the forecasting. $p = 3$ and $p = 26$ are found to viable candidates; however, only $p = 26$ turns out to give a model complex enough to capture the fluctuations and is hence chosen. The technique is implemented using the `statsmodel` module `VAR`.

4.4 Results

The performance of the VAR forecasting model was checked using root mean squared error (RMSE) - $RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_i - F_i)^2}$ - and the mean absolute Percentage error (MAPE) - $MAPE = \frac{1}{k}\sum_{i=1}^{k}\frac{|y_i - F_i|}{|y_i| + \epsilon}$, where ϵ is a small constant to prevent division by zero - for each feature [2].

The RMSE is low (< 10) for certain features for absolute pollutant concentrations - such as CO(GT) - and environment conditions - like humidity - but is somewhat high (~100) for certain metal detected features - such as (PT08.S5(O3)).

The MAPE values are acceptable (less than about 30%) for most pollutants, like non-metanic hydrocarbons and oxides of nitrogen, but show a high degree of error in temperature prediction. This is due to the effect of low temperatures (European cities often have temperatures very low in magnitude).

Since the test data, even at its present proportion, is a long time series; we expect our predictions to diverge more and more as we proceed further along the time axis, visualized in Figure 4.2.

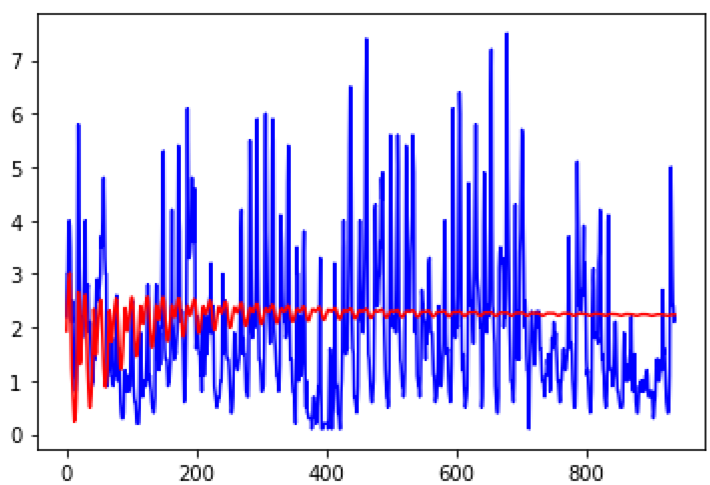

Figure 4.3: Actual values (purple) and predicted values (red) for CO(GT). Divergence increases as we proceed with time due to the large length of time series. Image Source: *See VectorAutoregression.ipynb.*

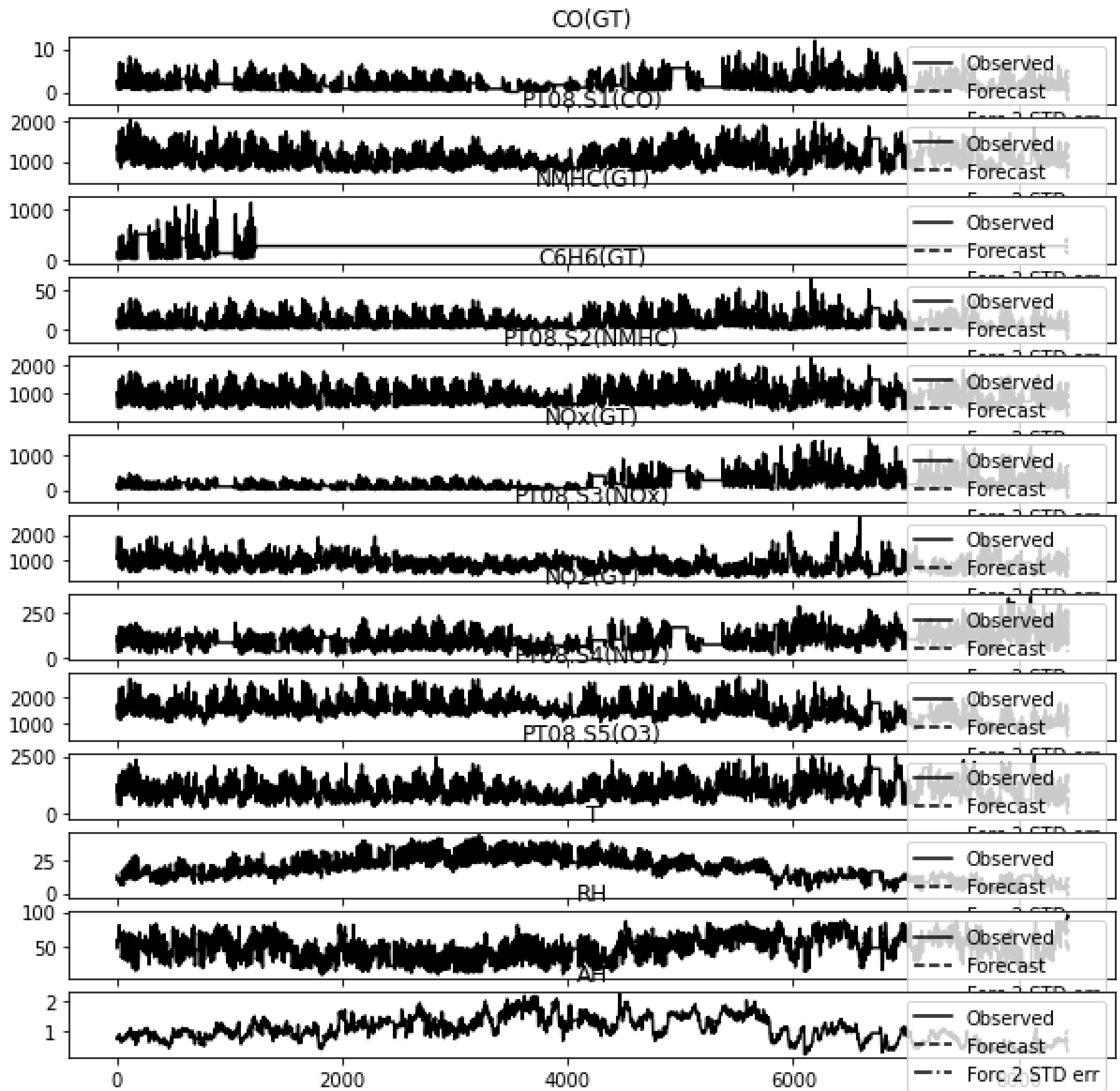

Figure 4.4 Forecast plot generated by vector autoregression.
Image Source: *See VectorAutoregression.ipynb.*

Please refer to the notebook VectorAutoregression.ipynb for all error values and plots.

References

[1]"Joe Kaeser: Data is the 21st century's oil, says Siemens CEO Joe Kaeser," *The Economic Times*. [Online]. Available: https://economictimes.indiatimes.com/magazines/panache/data-is-the-21st-centurys-oil-says-siemens-ceo-joe-kaeser/articleshow/64298125.cms?from=mdr. [Accessed: 04-Dec-2021].

[2] Anderson D., Sweeney D.J., Williams T.A., Camm J.D., Cochran J.J. (2014). Statistics for Business and Economics, 13th Edition, Cengage Learning. ISBN: 93-5350-251-9.

[3] "Time Series," *Wikipedia*, 30-Nov-2021. [Online]. Available: https://en.wikipedia.org/wiki/Time_series. [Accessed: 04-Dec-2021].

[4] "Time series forecasting: Definition, applications, and examples," *Tableau*. [Online]. Available: https://www.tableau.com/learn/articles/time-series-forecasting. [Accessed: 04-Dec-2021].

[5] Tan, P.-N., Steinbach, M.,, Kumar, V. (2005). Introduction to Data Mining. Addison Wesley. ISBN: 0321321367.

[6] Bengio, Y. & Simard, Patrice & Frasconi, Paolo. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council. 5. 157-66. 10.1109/72.279181.

[7] "Dynamic time warping clustering," *Cross Validated*, 01-Feb-1963. [Online]. Available: https://stats.stackexchange.com/questions/131281/dynamic-time-warping-clustering/131337#131337. [Accessed: 04-Dec-2021].

[8] S. Gharghabi, Y. Ding, C. M. Yeh, K. Kamgar, L. Ulanova and E. Keogh, "Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels," *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 117-126, doi: 10.1109/ICDM.2017.21.

[9] Rose S.L., Kumar L.A., Renuka K. (2019) Deep Learning Using Python. Wiley. ISBN: 9788126579914.

[10] UCI Machine Learning Repository: Character trajectories data set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Character+Trajectories. [Accessed: 04-Dec-2021].

[11] Ruiz, A.P., Flynn, M., Large, J. et al. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Disc 35, 401–449 (2021). https://doi.org/10.1007/s10618-020-00727-3.

[12] "PLASTICC astronomical classification," *Kaggle*. [Online]. Available: https://www.kaggle.com/c/PLAsTiCC-2018/data. [Accessed: 04-Dec-2021].

[13] "RAOTBL6: Data Set used by Yash P. Mehra (1994) in Urca: Unit root and cointegration tests for time series data," 02-May-2019. [Online]. Available: https://rdrr.io/cran/urca/man/Raotbl6.html. [Accessed: 04-Dec-2021].

[14] UCI Machine Learning Repository: Robot Execution Failures Data Set. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Robot+Execution+Failures. [Accessed: 05-Dec-2021].

[15] K. Kim and M. Choi, "Android malware detection using multivariate time-series technique," 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2015, pp. 198-202, doi: 10.1109/APNOMS.2015.7275426.

[16] Angryk, R.A., Martens, P.C., Aydin, B. et al. Multivariate time series dataset for space weather data analytics. Sci Data 7, 227 (2020). https://doi.org/10.1038/s41597-020-0548-x

[17] "Activity recognition," *Wikipedia*, 07-Nov-2021. [Online]. Available: https://en.wikipedia.org/wiki/Activity_recognition. [Accessed: 05-Dec-2021].

[18] A. Amidon, "MiniRocket: Fast(er) and accurate time series classification," *Medium*, 27-Aug-2021. [Online]. Available: https://towardsdatascience.com/minirocket-fast-er-and-accurate-time-series-classification-cdacca2dcbfa. [Accessed: 05-Dec-2021].

[19] "Sklearn.linear_model.RIDGECLASSIFIERCV," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifierCV.html. [Accessed: 05-Dec-2021].

[20] "Vector autoregression," *Wikipedia*, 13-Oct-2021. [Online]. Available: https://en.wikipedia.org/wiki/Vector_autoregression. [Accessed: 05-Dec-2021].

Datasets Used

Human Activity Recognition: The dataset "Human Activity Recognition using Smartphones" can be obtained at the link: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones.

Air Quality Forecast: The dataset "Air Quality" is available at the link: https://archive.ics.uci.edu/ml/datasets/Air+Quality. *Same data loading procedure as above.*