TIME-SERIES FORECASTING USING FEATURE BASED HYBRID APPROACH

by

Olashile S.Adebimpe

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2019

© Copyright by Olashile S.Adebimpe, 2019

To my mentor, friends, and family, I couldn't have done this without you.
Thank you for all your support along the way.

Table of Contents

List of Tables

List of Figures

Abstract

Machine Learning (ML) methods have been gaining prominence over time as interest in Artificial intelligence (AI) has been rising. Researchers have tried to explore these methods for time series forecasting models, moving away from the traditional statistical methods, but forecasting results from these models are still not impressive enough compared to traditional statistical methods in terms of forecasting accuracy and also the computational requirements.

In this research, we explored the unique strengths of both traditional statistical method and machine learning algorithms to propose a hybrid forecasting system based on a decomposition approach, with the objective of improving forecast accuracy across multiple forecasting horizons. Our proposed methodology uses the Seasonal and Trend decomposition using Loess (STL) decomposition procedure to break down time series data into trend-cycle, seasonal and covariance stationary components, where these components produce individual forecasts and these forecasts are aggregated back to whole using an aggregation procedure, with the sole purpose of minimizing errors. Various types of Exponential Smoothing algorithms were employed for the trend-cycle, seasonal components because of its unique weight combination approach while vectors of features were extracted automatically from the nonlinear covariance stationary subseries to create appropriate Machine Learning models.

We carried out our research using NN3 dataset and a large subset of 48,000 real-life monthly time series used in the M4 competition, which is characterized by considerable seasonality, trend and a fair amount of randomness so as to cover a wide range of time series structures. Our result reveals that the combination of decomposition, Exponential smoothing, Machine learning methods, and feature extraction gives less forecasting errors when compared to other combinatory approach and benchmark classical approach.

Glossary

AI	Artificial intelligence.
ANN	Artificial Neural Networks.
AR	Autoregressive.
GAs	Genetic Algorithms.
LS-SVM	Least Square Support Vector Machine.
MA	Moving Average.
MAPE	Mean Absolute Percentage Error.
MASE	Mean Absolute Scale Error.
MCP	Electricity Market Clearing Price.
ML	Machine Learning.
NN	Neural Network.
OWA	Overall Weighted Average.
SM	Statistical Methods.
sMAPE	Symmetric Mean Absolute Percentage Error.
STL	Seasonal and Trend decomposition using Loess.
SVM	Support Vector Machine.

Acknowledgements

I would like to express my special appreciation and gratitude to my supervisors Prof. Stan Matwin and Dr. Rita Orji for their useful comments, remarks, and engagement through the learning process of this master thesis. They have been a tremendous mentor to me and I would like to thank them for their support and advise on both research and my career goals. They consistently allowed this paper to be my work but steered me in the right direction whenever they thought I needed it.

I consider myself blessed to have got an opportunity to work under the direction of Dr. Stan Matwin and Dr. Rita Orji. I am ever grateful to Dr. Stan Matwin and The Institute for Big Data Analytics for funding and supporting my entire master's program.

I am grateful to my instructors – Dr. Stan Matwin, Dr. Vlado Kesselj, and Dr. Fernando Paulovich – for teaching Machine Learning for Big Data, Natural Language Processing, and Visual Analytics so well! The ideas gained from these classes have played a huge role in shaping my thoughts and have indirectly contributed to my thesis.

I would especially like to thank the students of the Institute of Big Data Analytics and Persuasive Computing group. All of you have been very supportive with your feedback, brilliant comments, and suggestions.

Finally, I must express my very profound gratitude to my parents and siblings, whose love and guidance are with me in whatever I pursue. I wish to thank my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Chapter 1

Introduction

Time series is defined as a collection of observations x_t made sequentially at specific time. Various examples of this kind of data occur in a variety of fields, ranging from economics to engineering, and the method of analyzing time series constitute an important area of statistics [1]. Time series data are known to be univariate records of only one variable, or multivariate - records of multiple variables in nature. On the other hand, these data could also exist in a continuous or discrete data form. In discrete form, the time series contains observations that are measured at successive discrete points of time with uniformly spaced intervals, while they are measured at some time intervals in continuous form. [2].

In discrete time series, the variables observed are assumed to be measured as a continuous variable using the real number scale. In different circumstances, continuous time series are easily transformed to discrete by merging data together over a specified time interval. This direction provided the foundation for this research work to focus on discrete time series, as it is known that the observations are recorded at equally spaced time intervals such as hourly, daily, weekly, monthly or even yearly.

Examples of time series are the variations of a stock index, sales figures, the annual flow volume of the River Nile at Aswan, the consumption of a certain goods, the daily blood pressure of an individual, daily air temperature or monthly precipitation in a specific location, the number of web page visits per second, the brain electrical activity of a patient measured at 256 Hz in an electroencephalogram (EEG), or innumerable other sequences based on industrial, economic, social phenomena, studies in medicine, geophysics, and engineering. Other types of data such as DNA or videos are not time series in their raw format, but they can be converted to time series. This transformation enables the use of a large number of algorithms, specially tailored to time series on other types of data.

1.1 Introduction to Time Series Analysis

Time series analysis provides methods useful for extracting potentially useful information from time-series data. These methods either help discover the characteristics of a physical system that generate the time series, to predict the changes of a time series, and to improve controls over the physical system. These time series analysis objectives are mostly classified as description, explanation, predictions, and controls. Description provides the simple properties of a times series data which include the trends or periodicity, seasonal variation, sudden or gradual changes. Explanation describes the variations between two or more related time series. Control is used to collect and analyzed appropriate corrective measures for a series of data. Lastly, prediction or forecasting provides future values of the time series using the observed past values and explained by [3] in *Smoothing, Forecasting, and Prediction of Discrete Time Series* where they used prediction to describe subjective methods and forecasting to describe objective methods. All these objectives require identification of a time series model, which interprets the series behavior and uses it for forecasting future values.

The process of discovering valid patterns and extracting useful information from a large amount of data using Machine Learning (ML) methods and statistical approaches is called Data Mining. With the amount of data being generated every microsecond, data mining mostly involves a large amount of data which are mostly referred to as Big data, available and mined for several reasons ranging from classification, clustering, and prediction or forecasting. The main topic of this thesis focuses on forecasting and is described as a broad research area concerned with the estimation of future events or conditions. Forecasting has been known to be very important in various fields and useful for various reasons, ranging from forecasting of stock prices and exchange rate in finance to forecasting variables like gross national product or unemployment in macroeconomics, or even used by companies to forecast the demand for their products to support planning and decision-making.

Forecasting applications are becoming widespread across various domains and have urged researchers, practitioners, educators, and decision makers to carry out ground-breaking research that improves forecasting knowledge. These research experiments constantly improve the accuracy of these forecasting models by testing and comparing with multiple reasonable methods. These have most recently prompted researchers to begin using more of evidence-based checklist for forecasting [4] as guidelines, similar to that used by other

practitioners in other fields likes medicine, aeronautics, and engineering [5]. This evidence-based check is meant to guide researchers against proposing forecasting methodologies without strong objective evidence regarding their relative performance over other standard forecasting tools, claiming methodology superiority with methodology inadequacies ranging from conclusions based on few or even single time series, and method evaluation on only a step ahead forecasting, and not comparing methods with benchmark methodologies.

Exhibit 1: Forecasting Methods Application Checklist

Name of forecasting problem: _____

Forecaster: _____ **Date:** _____

Method	Knowledge needed		Usable method	Variations within components
	Forecaster*	**Respondents/Experts†**	(☒)	(Number)
Judgmental methods				
1. Prediction markets	Survey/market design	Domain; Problem	☐	[ ]
2. Multiplicative decomposition	Domain; Structural relationships	Domain	☐	[ ]
3. Intentions surveys	Survey design	Own plans/behavior	☐	[ ]
4. Expectations surveys	Survey design	Others' behavior	☐	[ ]
5. Expert surveys (Delphi, *etc.*)	Survey design	Domain	☐	[ ]
6. Simulated interaction	Survey/experimental design	Normal human responses	☐	[ ]
7. Structured analogies	Survey design	Analogous events	☐	[ ]
8. Experimentation	Experimental design	Normal human responses	☐	[ ]
9. Expert systems	Survey design	Domain	☐	[ ]
Quantitative methods *(Judgmental inputs sometimes required)*				
10. Extrapolation	Time-series methods; Data	n/a	☐	[ ]
11. Rule-based forecasting	Causality; Time-series methods	Domain	☐	[ ]
12. Judgmental bootstrapping	Survey/Experimental design	Domain	☐	[ ]
13. Segmentation	Causality; Data	Domain	☐	[ ]
14. Simple regression	Causality; Data	Domain	☐	[ ]
15. Knowledge models	Cumulative causal knowledge	Domain	☐	[ ]
16. Combining forecasts from a single method… ☐		**SUM of VARIATIONS**		**[ ]**
17. Combining forecasts from several methods… ☐		**COUNT of METHODS**	**[ ]**	

**Forecasters must always know about the forecasting problem, which may require consulting with the forecast client and domain experts, and consulting the research literature.*

†Experts who are consulted by the forecaster about their domain knowledge should be aware of relevant findings from experiments. Failing that, the forecaster is responsible for obtaining that knowledge.

Figure 1.1: Forecasting methods application checklist [1]

Testing multiple hypotheses by comparing the accuracy of forecast methods with the accuracy of the forecast from currently used methods or benchmark methods are core requirements of a scientific approach in forecasting [6]. Combining this approach with methods guided by evidence-based forecasting approach which are consistent with forecasting principles have been shown to provide out-of-sample[2] forecasts with superior accuracy and

[1] Source: *http://forecastingprinciples.com/index.php/selection-tree*

[2] Out-of-sample forecast is similar to the test set in machine learning algorithm. It is used to formally evaluate the predictive capability of a model developed by forecasting observations that were not part of the data sample used to build the model

are always recommended [7].

This thesis focuses on ways of combining the unique strengths of several evidence-based forecasting to improve out-of-sample point forecasts accuracy because it has been established that it is unlikely for a single technique to consistently outperform to a great extent competitor method across a large number of series. We, therefore, focus on combining techniques from judgemental and quantitative methods to improve out-of-sample point forecasts accuracy. Example of these judgemental and quantitative methods are decomposition method and forecast combination, respectively, and Figure 1.1 shows a visual representation of the forecasting methods checklist.

1.2 Goals

The major objective of this thesis focuses on combining decomposition and forecast combination methods to improve forecasting accuracy across multiple forecasting horizons. Decomposition is achieved by making the best use of knowledge present in these series of data to break down a forecasting problem into several parts, forecasting each part separately, and combining the forecast of these different parts into a whole. And on the other hand, combining Statistical Methods (SM) and Machine Learning (ML) approaches to create a combination of methods where approaches in machine learning, statistical techniques, feature extraction, and many other methods are used for error minimization, thereby, significantly improving the out of sample forecast accuracy of our forecasting algorithm. The combinations of these two techniques or as most call it "*A Hybrid Approach*" is based on various assumptions that the data from the initial decomposition approach and the forecast produced from this hybrid methods are intended to be better than forecast from single individual methods.

For example, our proposed hybrid model could be useful for financial analysts to forecast the behavior of stock prices for economic profit by breaking down the historical data of the behavior of stock prices into several parts to understand the historical component of the series, producing individual forecast on these sub series and merging them back to form a whole series. The objectives of this research are to contribute to a better understanding about ways of exploiting findings and discovering patterns that enrich our understanding of ways decomposition and combination methods can help improve forecasting accuracy and the other factors that may affect it. At a general level, the following questions are presented

in order to investigate and clarify the above gaps.

- How does the performance of our proposed hybrid methodology compare to the other individual state-of-the-art classical approaches?
- How effective are Machine Learning models in hybrid methodology for time series forecasting?
- Does the decomposition of combinational approach compare to the other combination techniques like pooling, average and weighted average?

1.3 Contribution

The research summary and contributions of this thesis can be summarized as follows:

1. We present an approach to creating hybrid forecasting method using a simple and linear combinatory method. Most hybrid methods are implemented using a complex combinatory method via estimation of combination weights and this often introduces more error into the hybrid system which overwhelms intended gains expected.
2. We propose an approach that leverages on the unique combination of various methods such as Decomposition techniques, transformation, statistical methods, and machine learning methods to improve the forecasting power of a time series forecasting model. Our contribution highlights how these decomposition and transformation techniques passively help to improve the accuracy of point forecast in their own minute way.
3. We provide a scalable hybrid methodology that improves times series forecasting by effectively combining statistical techniques with machine learning methods. We show effective ways in which the modeling ability of machine learning methods could be harnessed in combination with statistical methods to improve the forecasting accuracy of hybrid models.
4. Lastly, a paper "*Improving the Accuracy of Time series Forecasting using Hybrid Approach*"was submitted and accepted for a presentation at the 39th International Symposium on Forecasting 2019 in Thessaloniki Greece.

Chapter 2

Introduction to Time Series Forecasting

Time series forecasting has always been a part of the temporal data mining functionalities alongside with classification, clustering and outlier detection. Time series forecasting of a set of time-ordered time series observation is the predictive task that involves the analysis of the time series data in order to predict future unknown changes or future values of the given observation. It involves the use of measured variables as potential predictors of future values of the target observations. The general assumption is that there is an unknown function that "maps" the past observations $y_1, y_2, ..., y_t$ into the future values y_{t+h} where y_i is the value of Y measured at time time i and h is the forecast horizon as shown in Figure 2.1. i.e. $y_{t+h} = f(< DescriptorOfThePast >)$ and the learning goal is to approximate this function using some prediction error criterion and historical record of the observed values [8].

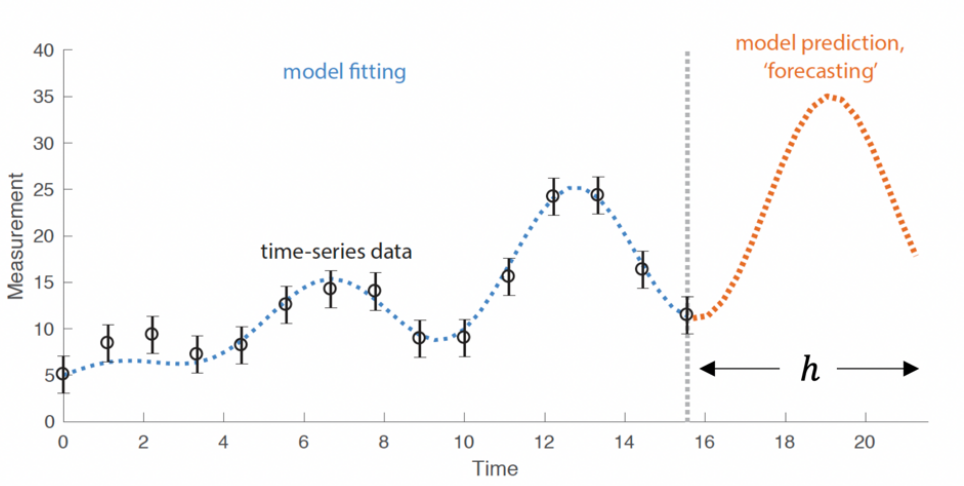

Figure 2.1: Time-series modeling and forecasting. The figure shows a uniform sampled time series(black), a model fit(blue), and out of sample forecast h of the fitted model(orange).

In general, time series forecasting describes a broad research method concerned with the estimation of future events similar to other methods in data analysis, ranging from methods like classification, clustering, and regression. We create models using timed data

to estimate future out-of-sample period, which is used for forecasting performance, comparatively to classification task that attempts to predict, for each individual in a population, to which class does this individual belong to. According to [9], forecasting approaches can be subdivided into two categories, which are qualitative models and quantitative models.

- Qualitative models, the models assume sufficient knowledge of an underlying process which is often based on experts judgment, subjective belief, and intuition that cannot be quantified.

- Quantitative models involve the automatic prediction of numerical data.

As briefly explained in Chapter 1, our main focus of this research is to find ways of improved methods of mining univariate discrete time series, which are modeled based on quantitative forecasting automatic prediction of numeric data using various methods, where data are being investigated for regularities and patterns in their past to extract knowledge that can help to predict the future data [10].

Time series forecasting can be classified based on forecast horizon h as short-term forecasting when the forecasting period is a short period of time usually less than three points or long time forecasting when the forecasting period is above twelve points and lastly, mid-term forecasting focuses on periods between the short and the long-term periods.

2.1 Time Series Forecasting Techniques

This section looks at forecasting techniques from a broad technique category point of view, ranging from Statistical Method (SM) to Machine Learning (ML) and Neural Network (NN) technique and lastly to the Fuzzy-based technique, which are mostly used for observations with more than one attribute. The literature review focuses on technique categorization instead of individual forecasting algorithm since all the forecasting algorithms can all be properly categorized into the above categories, and as there are variant of individual techniques used to different research to suit different reasons. We considered and reviewed three popular categorizations of different approaches for time series forecasting, discussing the contributions and publication on genuinely new forecasting algorithms. These categorizations are

1. Traditional or Classical Forecasting Techniques.

2. Stochastic or Machine Learning Forecasting Techniques.

3. Fuzzy Based Forecasting Techniques.

Most of these forecasting techniques have been used for various applications in numerous practical fields such as business, economics, science, engineering and many more [11][12]. These techniques involve appropriate model fitting that represents the underlying time series data. They could either be used individually by combining various techniques to create a hybrid model, or even using an optimization algorithm alongside with the forecasting model [13] to develop an efficient model with improved forecasting accuracy.

2.2 Traditional or Classical Forecasting Techniques

Out of the numerous approaches employed for time series forecasting, traditional techniques are known to be very popular and frequently used. The basic assumption about this approach is that the underlying time series is linear, has a particular statistical property such as normal distribution and that they are based mathematical formula and techniques, which are improved using several tools and automation methods. There are two widely used approaches, Autoregressive (AR) and Moving Average (MA) [14][15] and the combination of these two popular models give birth to more models like Autoregressive Moving Average (glsARMA) and Autoregressive Integrated Moving Averages (ARIMA) [16]. In this review, we will be discussing regressive models and Exponential Smoothing models, giving real-life examples of their application to the problem of time series forecasting.

2.2.1 Regressive Models

These models have drawn so much attention because of their relative simplicity in understanding and implementation. This method is a statistical method mostly implemented for predicting future response based on types of relationship between the data to be predicted, response history and the other factors that it depends on. There are different variables involves in regressive models. The dependent variable the main variable that we are trying to predict its values and the independent variables- whose influence and factor may affect the dependent variables.

Regressive models depend on the representation of various conditions used for predicting the future data and methods implemented to calculate these factors. These factors uses

historical data that helps understand the data and predicts values that closely represents the behavior of the dynamic system from which the experimental or observational data were gotten. Time series regression models are commonly used for modeling and forecasting of economic, financial, and biological systems. Various kinds of regressive models exist, but for the purpose of this review, we will be limiting our discussion on linear and multiple regression.

Linear Regression establishes the relationship between two variables using a straight line, by drawing a line that comes closest to the data by finding the slope and intercept that define the line and minimize the regression errors. The research of [17] explained the electricity load forecasting problem as one of the numerous forecasting problems where researchers try to predict the magnitude of the electricity load. The forecasting task is known to be a complex problem because of the changes and fluctuations in the electricity market. They stated that the load depends on numerous factors like the customer characteristics, the various days and the condition of the weather and utilized linear regression methods for this task.

The linear regression model requires several variables such as annual changes, the daily load changes and the seasonal changes. The transformation function was used to convert the previous year load data to future predictions using reflection and transformation. The linear regression mathematical model for this forecasting task is given below.

$$L(t) = L_n(t) + \sum a_i x_i(t) + e(t) \tag{2.1}$$

$L(t)$ - normal load at time t

a_i - estimated varying coefficient

$x_i(t)$ - independent factors that affect load

$e(t)$ - estimated varying coefficient

n - independent factors that affect load

Mbamalu and El-Hawary [54] computed regression coefficient using weighted least square for their proposed multiple regression method that predicted the electricity load and demands. In general, the mathematical model of the multiple regression model employed is given below

Regression coefficient were computed using weighted least square, these coefficients were used to propose a multiple regression method that predicted the electricity load and demands [18]. In general, the mathematical model of the multiple regression model employed is given below

$$Y_t = v_t a_t + \varepsilon_t \tag{2.2}$$

t - sampling time

Y_t - total load at time t

v_t - vector of variables that affect the load

a_t - vector of the regression coefficient

ε_t - model error

Due to their relative simplicity in understanding, regression models have drawn so much attention, but most practical time series shows non-linear patterns and several researchers, for example [19] presented a methodology how non linear models are appropriate for predicting volatility changes in economic and financial time series.

2.2.2 Exponential Smoothing

Exponential Smoothing (ES), a very known classical method used in many forecasting-based applications with an approach for smoothing time series data using the exponential window function. Exponential Smoothing methods originated in the 1950s and 1960s with the work of brown [3] and Holt [20], and have been widely used in business and industry domain.

In Optimal Properties of Exponentially Weighted Forecasts, Muth[21] suggested a statistical foundation for Simple Exponential Smoothing (SES) using exponentially weighted average that provides the optimal cast for a random walk plus noise , while the work of Pegels [22] used a simple classification of trends and seasonal patterns depending on whether they are additive linear or multiplicative nonlinear.

Exponential Smooothing model was proposed by Mogram and Rahman [23] using the functions given below and further steps towards putting exponential smoothing within a statistical framework provided by Box and Jenkins.

$$y(t) = \beta(t)^T f(t) + \varepsilon(t) \tag{2.3}$$

$f(t)$ - fitting function

$\beta(t)$ - coefficient vector

ε_t - white noise

Several works have been done by numerous researchers using Exponential Smoothing and in various contexts like computer components [24], air passengers [25], production planning [26] and electricity load forecasting [18].

Taxonomy that provides a useful categorization for describing the various methods of Exponential Smoothing was explained by [27]. Each method consists of one of five types of trends (none, additive, damped additive, multiplicative and damped multiplicative) and one of three types of seasonality (none, additive, and multiplicative). These 15 different simple ES methods are (no trend, no seasonality), Holts linear method (additive trend, no seasonality), Holt-Winters' additive method (additive trend, additive seasonality), and Holt-Winters' multiplicative method (additive trend, multiplicative seasonality) [28]. These various SES methods have been very useful in forecasting and also will be explored during the course of this thesis.

2.3 Stochastic or Machine Learning Forecasting Techniques

Time series data are not always linear as stated in the previous section and requires more complex approaches for data forecasting. The unique patterns in these series make forecasting difficult using linear regressive models, therefore, introducing stochastic models and computing based forecasting models, where models are built using one or more random variables or using Neural Networks. Future data are predicted using these types of models and in this research, we collectively regarded them as Machine Learning (ML) models. Stochastic modeling present data or predicts outcomes and estimates how probable outcomes are within a forecast for different situations. The Monte Carlo simulation, Support Vector Machine (SVM) are few out of the numerous Stochastic methods available.

Support Vector Machine and its numerous variants have been known to produce close forecast when compared to other forecasting methods such as Artificial Neural Network

and Bayesian Network, due to the proper selection of its hyper-parameters and problems such as overfitting, large out of sample types of data and local minimum problems can be avoided [13]. The SVM learning method is much simpler and easy to model and we will be considering SVM hybrid approaches and SVM with optimization approaches in this section.

Research by Xing Yan [29] compared the forecasting accuracy using Support Vector Machine (SVM) and the Least Square Support Vector Machine (LS-SVM) on the Electricity Market Clearing Price (MCP) forecasting dataset and the two methods showed several variations among which are that the LSSVM methods employ the equality constraints and a Sum-Squared Error (SSE) cost function while the traditional SVM uses a quadratic formula.

Xing also proposed a combinatory model of Support Vector Machine (SVM) and the Autoregressive Moving Average with External input (ARMAX) [11]. He compared the hybrid SVM-ARMAX model with existing single models such as single SVM, single LSSVM, single ARMAX and a hybrid LSSVM- ARMAX. The comparison showed that the hybrid SVM-ARMAX model is more accurate than other listed model because SVM models can obtain better forecasting accuracy by accumulating a linear module and its capability to handle outlier data.

Other methods uses optimization approaches to improve forecasting by combining Genetic Algorithms (GAs) optimization techniques with SVMs. H.Frohlich [28] used a wrapper method, depending on the learning algorithm based on SVM, to produce point forecast and these forecast results were based on ranking, while Genetic Algorithms (GAs) helped improve the ranking and parameter optimization.

Other methods use optimization approaches to improve forecasting by combining Genetic Algorithms (GAs) optimization techniques with SVMs. A wrapper method was used by H.Frohlich [30], depending on the learning algorithm based on SVM, to produce point forecast and these forecasts results were based on ranking, while Genetic Algorithms (GAs) helped improve the ranking and parameter optimization.

On the other hand, Artificial Neural Networks (ANN) and Neural Network (NN) have gained immersed popularity in the last few years, due to its successful results achieved in solving a huge amount of task ranging from different areas such as time series forecasting, pattern recognition, data clustering and classification. They are biological motivated and

are used to recognize regularities and patterns such as trends and seasonality in data, learn from experience and then provide generalized results based on previously known knowledge. The successes achieved by Neural Network models in the past few years are due to the fact that they are self adaptive, data-driven and nonlinear in nature, making them more practical and accurate in modeling complex data patterns as opposed to various traditional linear methods such as ARIMA [31][12].

Feedforward network (FNN) [12],Time Lagged Neural Network (TLNN)[32] and Seasonal Artificial Neural Networks [33] are some of the important NN models that have been used for forecasting problems while Probabilistic Neural Network (PNN) and Generalized Regression Neural Network (GRNN) [34] were used for classification and regression problems respectively. Neural Networks are amazing and powerful models used for forecasting but requires crucial network parameter selection and data transformation to produce an accurate forecast and in some cases face overfitting problems and issue of optimal network parameters determination.

In [35], Neural Network methods were employed to model quarterly time series prediction of trend and seasonal data while Temizel [36] provided a hybrid Neural Network technique using many random search algorithm and gradient search algorithms to improve the performance of seasonal time series prediction. In [37], they proposed a hybrid genetic optimization, where a gradient-based optimization was used for modeling the Radial Bias Function Network Based Autoregressive model (RBF-AR) on the United States of America(USA) census bureau data and the result was compared with the previously existing approaches such as SVM, ANN, ARIMA, and TDNN models

2.4 Fuzzy Based Forecasting Techniques

Fuzzy based techniques are mostly used for forecasting problems with more than one value of attributes or observation. This technique employs fuzzy sets for modeling and was developed by Song [38]. Fuzzy techniques comprise of three main phases which are the fuzzification stage, fuzzy rules, and relations definition stage and lastly the defuzzification stage. Fuzzy based time series prediction is most preferable for time series with complex pattern and is combined with other linear time series forecasting techniques.

A novel hybrid method was proposed by Egrioglu [39] where they combined the fuzzy

C-means clustering algorithm with the Artificial Neural Network (ANN) to solve the forecasting problem. The Fuzzy C means clustering algorithm was used to define the fuzzy rules and then fuzzy forecast was derived using the ANN.

	Model	Pros	Cons	Reseach
Traditional or Classical Methods	Linear Regression	Ability to handle different time series components and features. High interpretability.	Sensitive to outliers Strong assumptions.	Mbamalu (Mbamalu 1993)
	Exponential Smoothing	Ability to handle variable level,trend and seasonality components. Automated Optimization.	Sensitive to outliers Narrow confidence intervals	Brown (Brown 1959) Holts (Holt 2004)
	ARIMA (Auto Regressive Integrated Moving Avearge)	High interpretability. Realistic confidence intervals. Unbiased forecasts.	Requires more data Strong restrictions and assuptions Hard to automate	ARIMA(Zhang 2003)
Stochastic / Machine Learning Methods	Machine Learning Model	High Interpretability More transparent than other models Deals well with uncertainty. Control the variance of the components.	Higher holdout errors Higher training and evaluation time.	X. Yan(X. Yan 2013) X. Yan (X. Yan 2014) Frohlich(Frohlich 2004)
	Neural Network Model	Less restrictions and assumptions Ability to handle complex nonlinear patterns High Predictive power Can be easily automated	Low interpretability. Difficult to derive confidence intervals for the forcast. Requires more data. Overfitting of Data. Shortage of plenty training time. Optimal network parameters determination	P. G. Zhang (P. G. Zhang 2003) J. Faraway (J. Faraway 2008) C. Hamzaçebi(C. Hamzaçebi, 2008)
Fuzzy Based Methods	Fuzzy Based Models	Used for forecasting problems with more than one value of attributes Employes the concept of Fuzzy Logic.	Fuzzy rules are complex to set up	E. Egrioglu (E. Egrioglu 2013)

Table 2.1: Summary of various techniques used for time series forecasting with few advantages and disadvantages showing their strength and weakness.

These forecasting techniques categorization gives an overview of several methods that have been applied to forecasting problems in the past, but these techniques come with various strengths and weakness as seen in Table 2.1 , with a summary of each models pros and cons. It's been widely accepted that no universal single best method for solving these forms of forecasting problem "No Free Lunch Theorem". This has prompted researchers to focus on more techniques and methodologies to improve forecasting accuracy by combining various individual methods and harnessing their strengths to improve the forecasting model predictive accuracy. Forecasting competitions and discussion within the academic

forecasting community has helped open up new areas of academic research which has led to improved practice on valid and experimental designs [40].

On this note, we plan to explore the advantages of Machine learning methods and Statistical methods, to exploit their gains to improve forecasting accuracy by a careful combination of some of these methods. To forecast the behavior of future stock price, for example, we will examine the individual series and compare with methods gives best forecasting results and carefully combine them for our methodology gain as it has been established that Machine learning methods are used for complex and nonlinear methods while statistical methods are good for linear series.

2.5 Combinatory Methods

Results from accumulated researches over the past few years have continued to show the substantial improvement in forecasting accuracy as a result of combinatory approaches of different categories of forecasting techniques. For instance, linear data series achieve a high forecasting accuracy when a model with regressive models than when modeled with an Artificial Neural Network (ANN) or a Support Vector Machine (SVM). Also, some forecasting models tend to adapt well to a various component of time series data than others, thereby providing a good forecasting accuracy on these trended or seasonality data. A comprehensive research regarding the combination of forecasts [41] provided compelling evidence on the advantages of hybrid or combined technique forecasting. Combining approaches such as simple arithmetic average, weighted average and model switching are numerous techniques that have been employed for combinatory forecasting. These combinatory or hybrid approach have continued to show its superiority over individual techniques and this was also recently confirmed in the just concluded M-competition [53] where the best forecasting approach was a hybrid approach over a large number of time series data.The M-competition is a time series forecasting competition that enables researchers in the forecasting domain compare methodology with each other and learns how to improve forecasting accuracy and how these learning can be applied to advance the theory and practice of forecasting. The competition also compares techniques from experts with simple methods used as benchmarks. The major finding from the competition has always helped improve forecasting accuracy and directing research to methods that help improve forecasting accuracy. Introduction of hybrid models or combination of common

several methods and machine learning algorithms was confirmed to improve forecasting performance, which is one of the major conclusions from the last M4-competition [42] and likewise the NN3 [43] forecasting competition, which is a replica of the M3 competition [40] with an extension towards neural networks and computational intelligence methods.

Researchers have explored many combinatory approaches and results have shown that forecast from a combinatory method provides increased forecast accuracy compared to the individual methods [41]. Several combination techniques ranging for simple to complex are used, but simple methods are always advised and as stated by Timmerman [44]. Simple methods often outperform more complicated weighing schemes as errors are likely introduced through complex estimation of combination weights and, may overwhelm any gains from the setting the weight to their optimal values. Likewise, estimation errors from weight combination are known to be a serious problem for many combination techniques, especially when the sample size is small relative to the number of forecasts [45] [46] [47]. Combining forecast improves forecast accuracy majorly due to the diversification strength of each of the models. Also, models compliment each other against biases, measurement errors and loss functions. In this research, a simple form combinatory approach was used to improve on minimizing the error and more details about this form of combinatory approach are well explained later in this thesis.

2.6 Decomposition Methods

The main concept behind the proposed hybrid approach is the ability to capture all different patterns present in real-world time series data. We approached the combinatory method via the use of a decomposition function by breaking down time series into several components. We used several diverse methods to forecast each component separately and aggregating these individual forecasts back to a whole, making best use of individual data knowledge to ensure that relevant information is included in the forecast, leaving no valid reason for forecast adjustment. Our hybrid approach models the linear component of the series using well known statistical methods such as Exponential Smoothing variants while vectors of features are extracted from the complex patterns and modeled using Machine learning algorithms such as Gradient Boosting Algorithm and Support Vector Machine (SVM). This approach has been explored by various researchers and has led to an improvement in prediction performance as the nonlinear models overcome the limitations of the linear

models in modeling the nonlinear part of the time series data. The rest of this thesis provides detailed description and understanding of the various methods and approached we employed in the course of this research and is structured as follows: chapter three provides the methodology in details of our proposed model with a focus on the individual methods employed to create these hybrid model. chapter four provides explanation on our series of experiments and provide results and evaluation. Chapter five states clearly the challenges, our conclusion, and future plans.

Chapter 3

Time series Component

Time series components are defined as various factors that are responsible for bringing changes to the values of an observation in a time series data [48]. These components of variations are Trend (T), Cyclic Variations (C), Seasonal Variations (S) and Random or Irregular movements (I).

These components may be combined in different ways such as

Additive Model

$$y_t = T + S + C + I \tag{3.1}$$

Multiplicative Model

$$y_t = T \times S \times C \times I \tag{3.2}$$

Trend Components

Trends are long term change in the mean level of the time series data. They are observed to exhibits an increasing long-term pattern or decreasing long term pattern. If a time series does not show any increasing or decreasing pattern, then the series is regarded as stationary in the mean [48].

Seasonal Variation

Seasonal variations are short term movements that occur in data due to seasonal factor. They have the same or almost the same pattern of movement during a short period of time. Seasonality variation are present in a time series if the data are recorded hourly, daily, weekly, quarterly or monthly [48].

Cyclic Variation Components

Cyclic components are long term up and down movement around a specific trend. It is a kind of oscillations present in the time series and the duration are dependent on the type of business or industry being analyzed.

Irregular Components

These are the unpredictable component of time series that cannot be explained by trends, seasonal or cyclic movements. These variations are sometimes called residual or random components because their fluctuations are not systematic in nature and also show several unclear patterns. In time series forecasting, the objective is to model all the components to the point that the only component that remains unexplained is the random component [48].

3.1 Dataset

Two major types of datasets were used in our study to demonstrate the effectiveness of our proposed hybrid method. These datasets are the NN3 dataset [43] and the M4 monthly dataset [42]. They are from major forecasting competition events in the forecasting communities where the forecasting accuracy of various approach and methodology are compared with various benchmark models ranging from naive forecasting to advanced new statistical models and machine learning methods.

3.1.1 NN3 Dataset

The NN3 dataset contains 2 sets of datasets. Dataset A is a complete dataset of 111 different monthly time series drawn from a homogeneous population of empirical business time series. Dataset B is a subsample of 11-time series from the 111 series and is therefore contained in the larger dataset[1]. The subsample of 11 time series was used as validation dataset in the process of our experiments.

3.1.2 M4 Dataset

The M4 Competition Dataset consists of 100,000 time series data of yearly, quarterly, monthly and Other observation[2]. The dataset comes mainly from the Economic, Finance, Demographics, and Industry areas, while also including data from Tourism, Trade, Labor and Wage, Real Estate, Transportation, Natural Resources and the Environment. However, in this study, we only considered a subset of the M4 dataset, the 48,000 real-life monthly time series. We considered and used M4 and NN3 dataset in the scope of this research because it provides us numerous types and forms of data set in various lengths on which we

[1] http://www.neural-forecasting-competition.com/NN3/datasets.htm
[2] https://www.mcompetitions.unic.ac.cy/the-dataset/

can experiment with our proposed methodology. It also gives the opportunity to compare our methodology with other published methodologies, giving us valuable insight on how our proposed model performed relative to benchmark methods of these competitions.

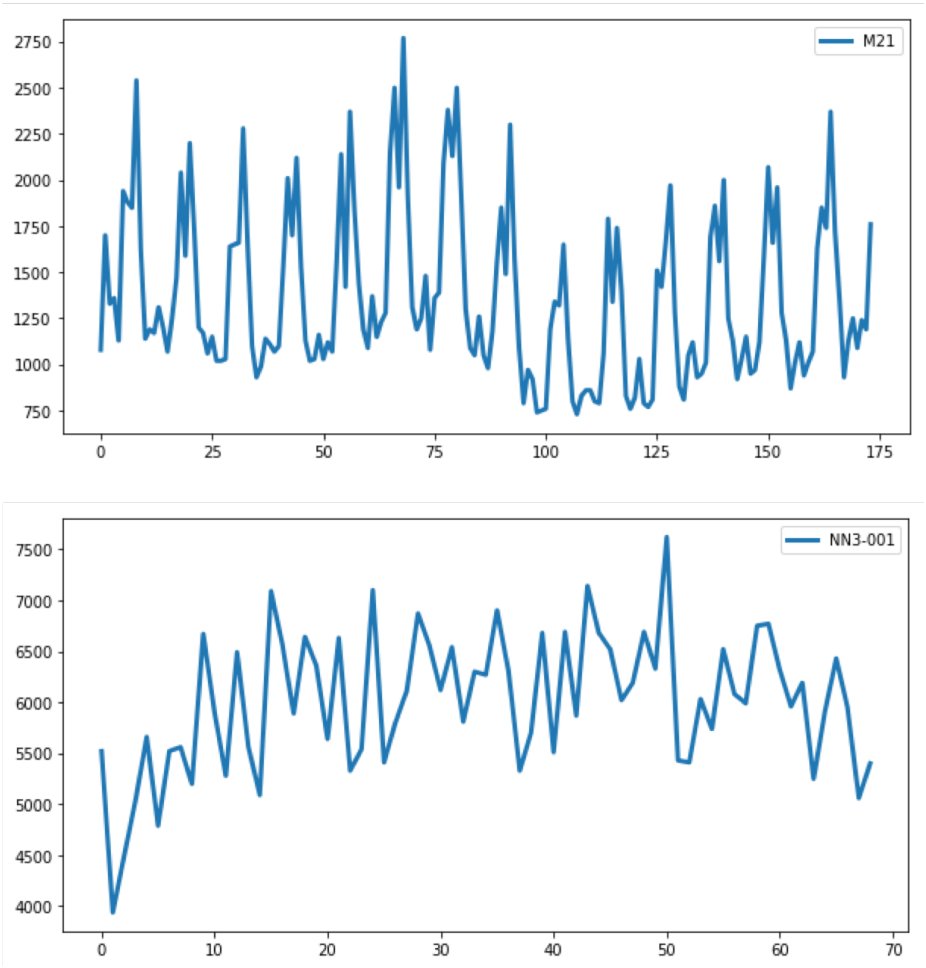

Figure 3.1: Image showing sample Time series data, M4 dataset(top) and NN3 dataset (bottom). The x-axis shows the range of values of the individual time series data and the y-axis is the number of observation (in months) available for modelling a forecasting model.

3.2 Methods

Several methods were employed for the various sections of our proposed hybrid methodology. In this section, we provided a detailed description of the different methods used in our proposed methodology as seen in Figure **??**. These methods include Data transformation, Data Decomposition, Statistical models and committee of Machine learning models.

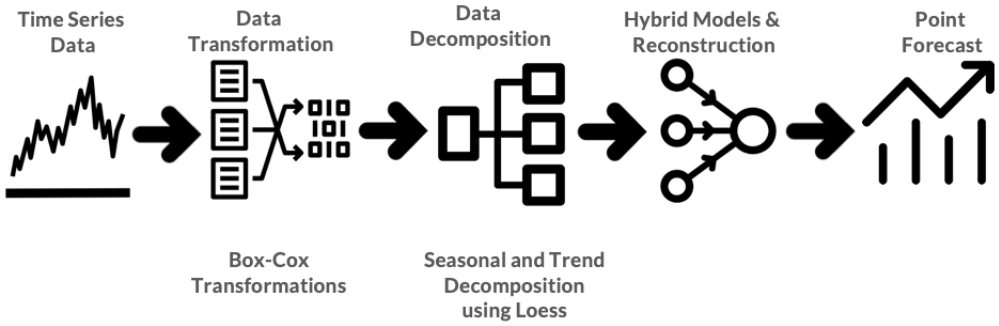

Figure 3.2: Image showing an overview of the various methods employed to create our hybrid model.

3.2.1 Data Transformation

Data transformation in the field of forecasting involve re scaling of historical data to simple patterns. Data transformation helps improve the accuracy of forecasting models because the transformation of a time series data leads to a stabilized variance series and helps make a linear relationship between the variable in a regression task. It also helps improve the fit of a forecasting model. Time series transformation could be scaling but in the context of this research, we carried out the transformation in terms of normalization by using mathematical transformation. The logarithmic transformation, square root transformation, and power transformations are few among the numerous mathematical transformation approaches employed by researchers.

The Box-Cox transformation is a popular and general class of transformation that was created by Box and Cox [49] in 1964 to stabilize the variance of a time series. It includes both logarithms and power transformation to remove heteroscedasticity (non-constant variance) of a variable and make the series look like more normally distributed. It depends on a parameter lambda λ and is defined as follows.

$$w_t = \begin{cases} \log\left(y_t\right), & \lambda = 0 \\ \left(y_t^{\lambda} - 1\right)/\lambda, & \lambda \neq 0 \end{cases} \tag{3.3}$$

This statistical technique is very useful for statisticians and economists regarding normality and homoscedasticity assumptions for linear models. A good value of lambda λ makes the size of seasonal variation constant across the whole series, thereby making the forecasting model simpler. But the difficulty of choosing an optimal lambda λ restricted

our intervals to be between [1,1]. To obtain the value of lambda λ, we followd the methods of [50] which divides the series into subseries of length equal to the seasonality, then the mean m and the standard deviation s are calculated and the lambda λ is chosen in such a way that the coefficient of variation $s/m^{(1-\lambda)}$ aross the subseries is minimized.

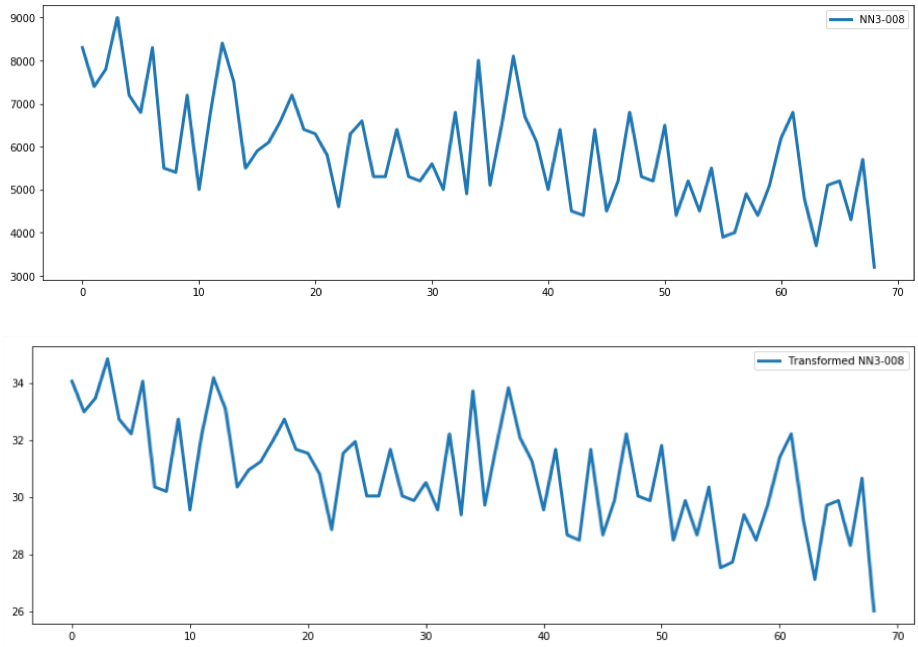

Figure 3.3: Original series(above) and the Box-Cox transformation version of NN3-008 time series from the NN3 dataset with lambda λ = 0.2494. The transformation function converted the range of values of the original NN3-008 series is transformed from a range of 3000-9000 to a range of 26 - 35

After a forecast has been generated from the transformed data, a reverse transformation is used to obtain the forecast on the original scale using the reverse Box-Cox transformation that is defined by

$$y_t = \begin{cases} \exp(w_t) & \text{if } \lambda = 0 \\ (\lambda w_t + 1)^{1/\lambda} & \text{otherwise} \end{cases} \tag{3.4}$$

Box Cox transformation was employed as a major transformation tool in this research because of its interpretable[3] nature due to its logarithms and power transformation. It is useful for time series data that grows exponentially and constrain forecast to stay positive on the original scale [51]. It also produces a normal distribution of the transformed data and

[3]The logarithm in Box-Cox transformation is always a natural logarithm. So, if lambda λ = 0 natural logarithm is used, but if lambda $\lambda \neq 0$, then a power transformation is used, followed by some scaling.

a constant variance which is valuable in removing spurious interactions and helps identify the factors that are really significant.

3.2.2 Decomposition

Time series decomposition is a statistical forecasting technique that decomposes historical data into various components, where each component is representing an underlying pattern. It helps extract various components of the series, forecast separately and combine thereby improving the understanding out the time series data and also the forecast accuracy. Classical decomposition, Fast Fourier Transformation (FFT), Discrete wavelet transform (DWT) and STL decomposition are popular known decomposition technique mostly used.

FFT uses spectral analysis to convert a time series data into a representation in the frequency domain [52] while DWT produces a time-frequency representation of a time series with a higher resolution of time [53]. These two approaches were flawed in our research because they lack the abstraction of breaking down series into systematic and unsystematic components. A given time series is thought to consist of three systematic components including level, trend, seasonality, and one non-systematic component called noise.

On the other hand, Classical and STL decomposition are relatively simple procedures of time series decomposition, that breaks down a series into components that actually make up the composition of the series itself. These components are Trend (T), Cycle (C), Seasonality (S) and Irregular movements (I) as explained earlier. Classical decomposition employs a simpler method to determine the different components which results in the trends and rapid rises and falls of the seasonal component being over-smoothed. It also assumes that the seasonal component repeats from year to year and the first few and last observation in the trend estimate are always unavailable. This is caused by using moving average filter to determine the trend component. These difficulty attributes of the classical decomposition make out of sample forecast tedious to compute. These flaws were improved in STL decomposition and thus provide more advanced functionality which is robust to outliers [51].

3.2.3 Seasonal Trend Decomposition using Loess

Seasonal and Trend decomposition using Loess smoother, mostly called STL decomposition method was developed by Cleveland, McRae, Terpenning [54]. It is a simple, versatile and robust method for decomposing time series with the help of a sequence of applications of Loess smoother. It uses loess interpolation to smoothen the cyclic sub-series to determine the seasonal component and to smooth out the estimated seasonal component In a final step, the de-seasonalized series is smoothed again to find an estimation of the trend component. This process is repeated several times to improve the accuracy of the estimations of the components.

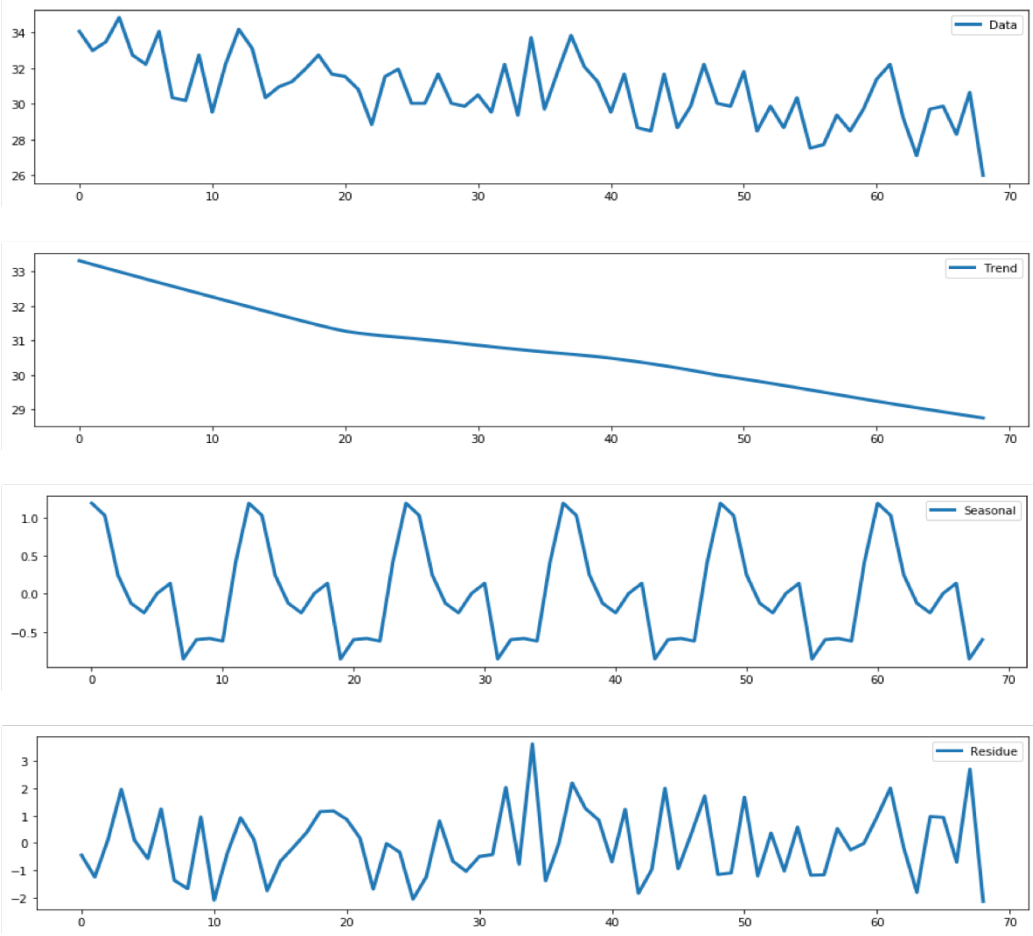

Figure 3.4: STL decomposition into trend, seasonal part and residue of the Box-Cox transformed version of series NN3-008 from the NN3 dataset.

The periodicity of the observed series, an important parameter, based on the kind of series being observed. Our experiment only made used of monthly data from NN3 and

M4 dataset, thereby defining our period parameter p = 12 for all series under consideration. STL decomposition of series into trend, seasonal and reminder are mostly additive in nature i.e. summing the parts give the original series again. Advantages of STL decomposition in this experiment over other decomposition method is that the seasonal component can change over time, robust to the effect of outliers on the calculation of trend and seasonal component and it includes all points of the time series data in the result of the STL decomposition. In the implementation, the trend component is calculated by substituting a configurable Loess regression for the convolutional method used in seasonal˙decompose. Figure 3.4 shows the STL decomposition of series NN3-008 from the NN3 dataset as an example.

3.2.4 Feature Extraction

In data mining and machine learning, feature extraction methods involve ways of looking for characteristics in data that help solve the given problem. It is defined as a process of creating new features from an initial set of data, where these features encapsulate the central properties of a data and represent it in a low dimensional space that facilitates the learning process. These extracted features include statistical features such as correlation structure, distribution, entropy, stationarity and scaling properties, which provides vector feature representation of the time series and also facilitates time series fit into a range of time series models. These features are mainly related to the statistical information of the data set.

Feature extraction can be performed using various time series analysis and the feature can be obtained by using several techniques such as data time feature, lag feature and rolling window features [55]. Data time features are the simplest form of feature. These features are extracted from the date and time of each observation and are generally known to produce poor model because they do not capture the statistical properties of the underlying time series data. Lag features are features generated from observation at prior time steps. However, the order of the observation in lag feature must always be preserved in order for the model to produce expected result.

Our research used the rolling or sliding window feature strategy for engineering feature. The statistical properties of the series are modeled by breaking the series into fixed length. A sliding window w is defined and the statistical informative feature is extracted from the

window. This information is then transferred into a feature vector that is incorporated into our proposed framework to produce a point forecast as explained in Figure 3.5. These features include various statistical properties such as mean, median, skewness, kurtosis, autocorrelation and many more.

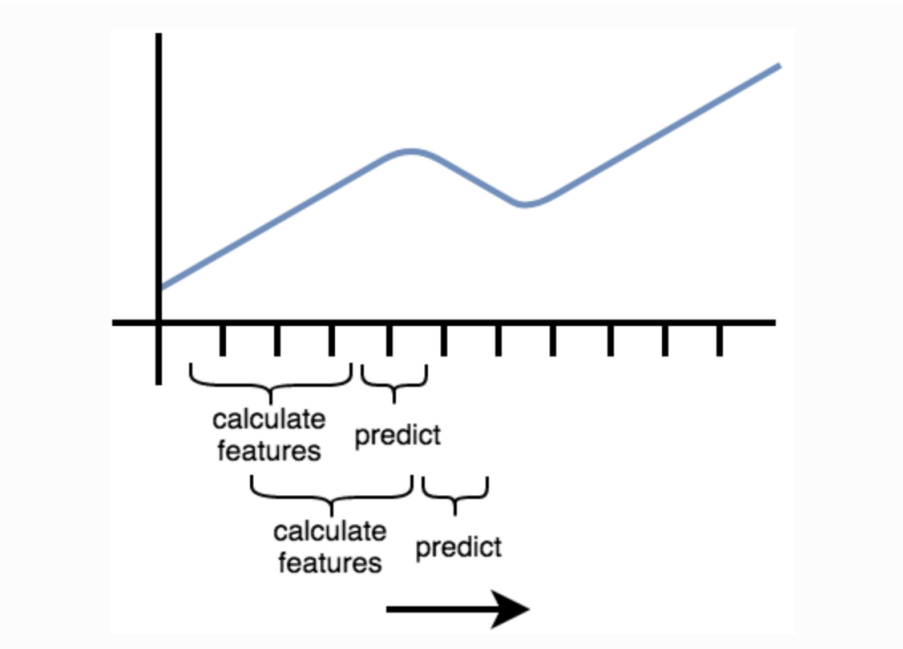

Figure 3.5: Image showing how the moving sliding window which extracts statistical features and then predict the next step forward (not included to extract feature)[4]

.

Table 3.1 shows a description and names of different time series features extracted to make point forecast.These statistical features of the residue components are used to train and build Machine learning methods that develop forecasting models used for predicting the next set of points.

3.2.5 Forecasting Models

The dynamic combination of several expert models have shown to provide a superior predictive performance relative to a single algorithm [56] and the work of [57] shows one of the numerous examples of combining these experts based on arbitrating.

Our hybrid methodology employs the combination of a diverse set of both statistical

[4]https://tsfresh.readthedocs.io/en/latest/text/forecasting.html

SN	Feature	Description
1	Mean	The average value of the series of data in the sliding windo
2	Standard Deviation	Measure of the spreadness of the data over the window
3	Variance	The square of standard deviation
4	Skewness	The degree of asymmetry of the series distribution
5	Kurtosis	The degree of peakedness of the series distribution
6	Mean change	The mean over the differences between subsequent time series values in the sliding window
7	Minimum	The highest value of the time series in the sliding window
8	Maximum	The lowest value of the time series in the sliding window
9	Mean of Absolute Change	The mean over the absolute differences between subsequent time series values in the sliding window
10	Linear Trend	Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one
11	Aggregation Autocorrelation	Calculates the value of an aggregation function e.g. the variance or the mean over the autocorrelation for different lags.
12	Autocorrelation	The autocorrelation of the specified lag
13	Pairwise Correlation	Correlation between two axes (channels) of each sensor and different sensors
14	Interquartile Range	Measure of the statistical dispersion, being equal to the difference between the 75th and the 25th percentiles of the signal over the window

Table 3.1: Classical time-series features that were extracted from the residual component of the time series data

methods and machine learning methods to make predictions. After which they are linearly combined together, based on the applied decomposition techniques. Statistical methods such as Holts Linear Trend [58] and Holt-Winters seasonal [59] methods are the two variants of Exponential Smoothing algorithm used to capture the trend and seasonality complexity of the series after being broken down into systematic components by STL decomposition. On the other hand, boosting algorithms such as XGBoost [60] and Gradient Boosting Machine [61], and Support Vector Regression (SVR) [62] were employed due to their ability to model complex structure of data, to fit the extracted features of the residual components from the time series data.

Statistical Models

- **Naïve Method**: These are forecasting technique that assumes that the next expected point is equal to the last observed point. Hence, the naïve method assumes that the most recent observation is the only important one, and all previous observations provide no information for the future.

$$\gamma_{t+1} = y_t \tag{3.5}$$

  The naïve method is mostly useful for stable datasets and is used in the evaluation metrics. As all methods are meant to better than forecast from naïve methods.

- **Exponential Smoothing**: Exponential smoothing algorithms have been one of the most frequently used forecasting techniques for numerous reasons. It is known for its incredible track record of success with minimal data requirement. The forecasting formula behind this awesome model is very simple to understand and only requires smoothing constant called the weighting factor, the forecast for the current period and the actual demand for the current period to predict the forecast for the next period. Forecasts are calculated using weighted averages where the weights decrease exponentially as observations come from further in the past, the smallest weights are associated with the oldest observations. The simplest version of Exponential Smoothing (ES) algorithm is the Simple Exponential Smoothing (SES) and is most suitable for data with no clear trend or seasonality pattern. SES formula is given as

$$S_{t+1} = \alpha y_t + (1 - \alpha) S_t \tag{3.6}$$

  Which can also be written as

$$S_{t+1} = S_t + \alpha \epsilon_t \tag{3.7}$$

  t $>$ 0,0 $\leq \alpha \leq 1$ is the smoothing parameter. ϵ_t is the forecast error for the period t.

- **Holts Linear Trend Method**: Holt Linear Trend Method[58] is similar to Simple Exponential Smoothing, but extending it to allow forecasting of data with a trend. It takes the trend level into account without any assumption. This method is improved by the introduction of a second equation with a second constant, γ , which is always

chosen in conjunction with α ,

$$\begin{aligned} \ell_t &= \alpha y_t + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right) \quad 0 \leq \alpha \leq 1 \\ b_t &= \gamma\left(\ell_t - \ell_{t-1}\right) + (1-\gamma) b_{t-1} \quad 0 \leq \gamma \leq 1 \end{aligned} \tag{3.8}$$

The first smoothing equation adjusts S_t directly for the trend of the previous period, b_{t-1} by adding it to the last smoothed value, S_{t-1}. This helps to eliminate the lag and brings S_t to the appropriate base of the current value.

Therefore, the forecast equation for the next value will be

$$S_{t+h} = \ell_t + h b_t \tag{3.9}$$

- **Holt-Winters seasonal method**: This statistical forecasting method was created by Holt [58] and Winter [59] by extending the Holts method to capture seasonality. It takes account both trend and seasonality by using three smoothing equation to forecast future values.

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t + h b_t + s_{t+h-m(k+1)} \\ \ell_t &= \alpha\left(y_t - s_{t-m}\right) + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right) \\ b_t &= \beta^*\left(\ell_t - \ell_{t-1}\right) + \left(1-\beta^*\right) b_{t-1} \\ s_t &= \gamma\left(y_t - \ell_{t-1} - b_{t-1}\right) + (1-\gamma) s_{t-m} \end{aligned} \tag{3.10}$$

Machine Learning Models

Numerous data series from NN3 and M4 contains complex structure which seems very difficult to be modeled by a single Machine Learning (ML) method. Due to this, we created a committee of Machine Learning methods that models separately and select model with the lowest in-sample error from the complex structure of the residual component. Our committee of Machine Learning models contains Boosting Algorithm and Support Vector Regression algorithm, and they helped handled appropriately the intrinsic properties of the time series data based on the statistical feature extracted.

- **The Boosting Algorithm**: The XGBoost algorithm [60] is a decision tree- based ensemble Machine Learning algorithm that used a gradient boosting framework. Neural

Network has been proven multiple times to outperform all other algorithms when the problem is relating to data in its unstructured form [63]. However, when it comes to structured data, decision tree-based algorithm is considered best-in-class right now. XGBoost algorithm [60] has been used for a wide range of applications ranging from regression, classification and prediction problems. It is similar to the Gradient Boosting Machine (GBM) as both use ensemble tree method that apply the principle of boosting weak learning using gradient descent architecture. However, XGBoost improves GBM framework by providing optimization enhancement such as tree pruning and parallelization, as well as algorithms enhancement such as regularization.

- **Support Vector Regression (SVR)**: The Support Vector Regression [64] is a successful method based on using a high-dimensional feature space formed by transforming the original variables. It uses the same principle as the Support Vector Machine (SVM) for classification, with only a few minor differences has the output here as an infinite possibility. SVR uses SVM to try and identify the hyperplane that maximizes the margin between the classes and minimize the total error under tolerance. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. The prediction as described by [65] is given by

$$f(x) = \omega^T x + b \tag{3.11}$$

where, ω is the weight vector, b is the bias and x is the input vector. And the error function is given by

$$J = \frac{1}{2}\|\omega\|^2 + C\sum_{m=1}^{M} |y_m - f(x_m)|_\epsilon \tag{3.12}$$

where, x_m and y_m denotes respectively the mth training input vector and target output. We have the Linear SVR, Non-linear SVR and the kernel functions such as Gaussian kernel. Since our study in this research focuses on the accuracy rather than the complexity, forecasts were produces using a ϵ - regression SVM which maximizes the borders of the margin under suitable conditions to avoid outlier inclusion from the residue component, allowing the SVM to automatically decide the number

of support vectors needed. Gaussian Radial Basis Function (RBF) was used in training and predicting as it is generally known for its good general performance and few parameter requirements.

3.3 Evaluation Metrics

Various types of error metrics exist for time series forecasting problems, with these metrics having limitations, and if not used properly could leads to misleading results. In this section, we discussed four types of forecast error metrics, which are the absolute error-metric, the percentage error metric, the relative error metric, and the scaled free error metric. We discuss their shortcoming in relation to our proposed model and dataset used for testing, giving an appropriate reason for the error metric we chose for our research.

- **The Absolute Error Metrics**: Examples of the absolute error metric are Mean Absolute Error (MAE), Mean Square Error (MSE), Geometric Mean Absolute Error (GMAE). Absolute and squared in this type of error metric are used to prevent cases where negative and positive error offset each other. They are very easy to understand, compute and interpret and are best when computing the forecasting error on a single series. However, when measuring forecasting error across multiple series of data, the absolute error metric cannot be used because it is scale dependent.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \gamma_i|}{n} \tag{3.13}$$

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \gamma_i)^2}{n} \tag{3.14}$$

  where,
  y = Actual Forecast
  γ = Forecast

- **The Percentage Error Metric**: The Percentage error metric is also very easy to understand and interpret in terms of Percentage Better. It has the advantage of being immune to outliers, likewise being scale independent and can be used to compare forecast performance between different series. Examples of this kind of metric is

Mean Absolute Percentage Error (MAPE). Limitation of measurement based on percentage errors is that percentage values could be infinite or undefined if there are zero values in the series, and with the occurrence of zero periods of demands in intermittent-demand data, using percentage metric leads to confusing results.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \gamma_i|}{y_i} \tag{3.15}$$

The Symmetric MAPE (sMAPE) , which solves the problem of MAPE of putting a higher penalty on positive errors than on negative errors, was used in the M3 competition [40] . It is defined below and employed for error measuring metrics in this research.

$$sMAPE = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_i - \gamma_i|}{(|y_i| + |\gamma_i|)/2} \tag{3.16}$$

- **The Relative Error**: This scale dependent type of error allows us to compute error metrics by dividing each error obtain by error obtained from benchmark methods usually naïve method. The relative error is generally denoted by e_t/e_t^*, where e_t^* is the error obtained from the benchmark methods.Examples are Mean Relative Absolute Error (MRAE), Median Relative Absolute Error (MdRAE), and Geometric Mean Relative Absolute Error (GMRAE). One of the major issues with the relative error is that, in cases of intermittent demand data, errors will be small and will result in zero errors and division by zero.

- **The Scaled Free Error**: Mean Absolute Scale Error (MASE) was proposed by Hyndman and Koehler [66]. It is a scale-independent error metrics, generally accepted for comparing forecast accuracy across multiple series because of its scale-free nature. It is also very easy to understand and interpret as they scale error based on the in-sample MAE from the naïve methods, with values above less than one signifying a good forecast accuracy for one step ahead forecast horizon. The values may be larger than one in multiple forecast horizon.

$$MASE = \frac{1}{h} \frac{\sum_{t=1}^{h} \left| Y_t - \widehat{Y}_t \right|}{\frac{1}{n-m} \sum_{t=m+1}^{n} |Y_t - Y_{t-m}|} \tag{3.17}$$

MASE is widely used as one of the most accepted forecasting error metrics because of its scale independent nature and also used as a measuring metric in this thesis

alongside sMAPE. It can be used to measure forecasting accuracy on single series and across multiple series by averaging the MASE values on each series. Hyndman [66] stated that MASE is the only error metrics that can be used on forecasting situations with common origin, varying origin and forecast from multiple series. It is also more preferable than MAD/MEAN, as the latter assumes the series is stable over time which makes it unrealizable when the data exhibits trends, seasonality or other patterns.

3.4 Multiple Horizon Forecast

There are three major ways in which ML models are used to achieve multiple horizon forecast, and they are Iterative, Direct and Multi-Neural Network forecasting. In iterative forecasting approach, the first forecast is achieved the same as the one step ahead, where the training data up y_t are used to forecast y_{t+1} While for subsequent forecast, the previous forecasts are used instead of the actual value. For example, to the get the forecast for horizon 3, we use the forecast value for horizon 1 and 2, until the prediction of all the horizon. The direct forecasting approach is more complex and computationally demanding that the iterative approach as uses a Neural Network and require the single NN model to have h output nodes, which is used to produce all the h various forecast for each horizon simultaneously.

Multi-Neural Network forecasting approach involves separate NN models to produce multiple forecast horizon h, separate NN models are trained, where each one is used for predicting a single h step ahead forecast. Complex approaches of extrapolation through ML methods, such as direct and Multi NN approach has been explained to give less accurate results when compared to the iterative approach [67]. The advantage of the iterative approach, being very simple and computationally easy, made it our choice of multiple horizon forecast approach on the ML side of our hybrid approach even though the accuracy of the forecast deteriorates as the forecast horizon increases because the new forecast depends on the accuracy of the previous ones. In addition, the length of each series is not so long for us to consider using NN models as they require a lot of data and take a huge amount of time training.

3.5 The Overall Procedure

In this study, we used STL decomposition to obtain the systematic components of the series involved before mining to produce the forecast. Our proposed hybrid methodology as shown in detail in figure 8 consist of two major stages which are the decomposition and the forecast combination after the series has been transformed.

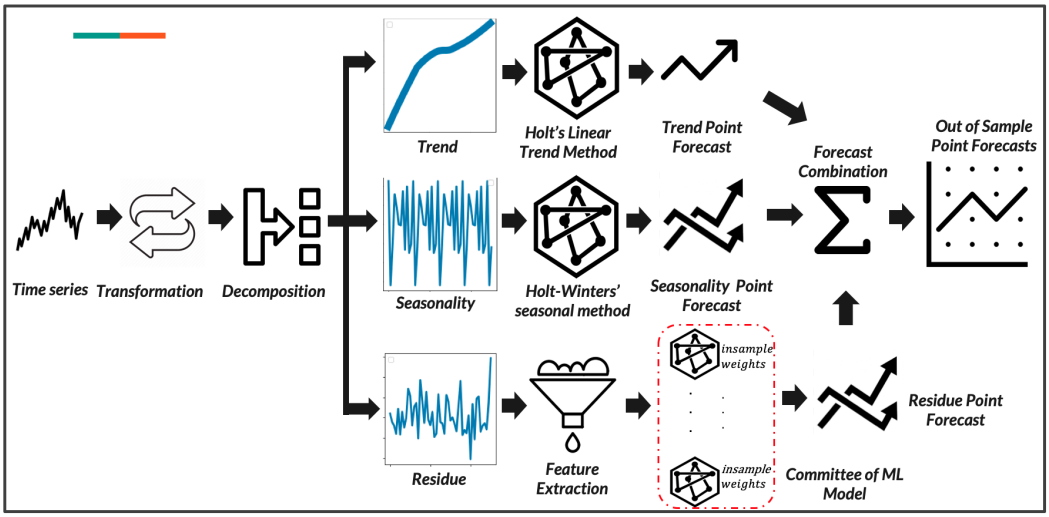

Figure 3.6: The Overall Procedure of Our Hybrid Methodology

In the first stage, after the series has been transformed, it is decomposed into trend, seasonality, and residue which are then mined individually in the second stage. Exponential smoothing models such as Holts linear and Holts winter methods are employed to produce out of sample point forecast for the trends and seasonality data, while due to the complex nature of the residue, statistical methods are not sufficient to detect the non-linear patterns, therefore set of statistical features listed in Table 3.1 are extracted from the residue subseries and Machine Learning algorithms such as Gradient Boosting, XGBoost and Support Vector Regression algorithm form a committee of models, which models the complex statistical properties of the residue component. A detailed representation of the algorithmic flow is explained below with a full detailed algorithmic flow in Algorithm 1 and Algorithm 2.

Algorithmic flow for our proposed hybrid methodology

Input: For time series $\{y_1, y_2, \ldots, y_t\}$

Output: $\widehat{y}_{t+h|t}$ where h is the forecast horizon

For a given series $\{y_1, y_2, \ldots, y_t\}$:

1. Transform the time series using the Box-Cox transformation using the optimal parameter λ to make the series y_t as normal as possible. Where the minimum of y_t is non-negative for all series used, we choose $\lambda \in (-1, 1)$ to minimize the Shapiro-Wilk statistics.

$$Y_t^* = f_\lambda (Y_t) \tag{3.18}$$

$$w_t = \begin{cases} \log (y_t), & \lambda = 0 \\ \left(y_t^\lambda - 1\right)/\lambda, & \lambda \neq 0 \end{cases} \tag{3.19}$$

2. Using STL decomposition, the transformed series Y_t^* is decomposed into its systematic components.

$$Y_t^* = T_t + S_t + R_t \tag{3.20}$$

Therefore, the de-trended data after Box-Cox transformation is

$$X_t = Y_t^* - T_t \tag{3.21}$$

While de-seasonalized data after Box-Cox transformation is

$$Z_t = Y_t^* - S_t \tag{3.22}$$

Time series after trend and seasonality adjustment is

$$Y_t' = Y_t^* - T_t - S_t \tag{3.23}$$

Where the measure of trend and seasonality are $1 - \mathrm{Var}\left(Y_t'\right)/\mathrm{Var}\left(Z_t\right)$ and $1 - \mathrm{Var}\left(Y_t'\right)/\mathrm{Var}\left(X_t\right)$

3. Using Holts linear trend method to obtain the trend forecast of the series.

$$\widehat{T}_{t+h|t} = \ell_t + hb_t \tag{3.24}$$

Where ℓ_t and b_t respectively are the level estimate of the trend and the slope at time t which are controlled by the smoothing parameter α and γ

4. Also, the Holt-Winters seasonal method is used to obtain the seasonality forecast.

$$\hat{S}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \tag{3.25}$$

   Where k ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample.

5. After a series of experiment, an optimal sliding window w and relevant statistical features list were determined, a vector of statistical features is extracted from the residue $\{r_{t-w}, \ldots r_t\}$, which helps in the prediction of $\hat{r}_{t+1}$

6. The model selection algorithms perform an in-sample forecast and assign weights to models to determine which machine learning model is best suitable for the time series at hand.

7. After returning the best model suitable, the next forecast point $\hat{r}_{t+1}$ is determined and merged back to the series $\{r_{t-w}, \ldots r_t, \hat{r}_{t+1}\}$ to determine $\hat{r}_{t+2}$. This process is repeated until we have all the point forecast until $\hat{r}_{t+h}$.

8. With the forecast of the three systematic components available, $\widehat{T}_{t+h|t}$, $\hat{S}_{t+h|t}$ and $\hat{r}_{t+h|t}$, and the knowledge that STL division is additive, we linearly combine these components to form a whole.

9. An inverse Box-cox transformation is used to obtain the forecast on the original scale $\hat{y}_{t+h|t} = f_\lambda^{-1}\left(\hat{T}_{t+h|t} + \hat{S}_{t+h|t} + \hat{r}_{t+h|t}\right)$.

Algorithm 1: Out of Sample Point Forecast Algorithm for predicting time series forecast points.

Input : Time series Y up to time t

Input : Window size w

Input : Forecast horizon h

Input : [FeatureList]

Output: $\hat{y}_{t+h}$

$\lambda \leftarrow Boxcox.Lambda(Y)$

$Y.bc \leftarrow Boxcox(Y, \lambda)$

if $(Y.bc$ *is seasonal)* **then**

  [trend, seasonality, remainder] $\leftarrow \mathrm{stl}(Y.bc)$

else

  seasonaliy $\leftarrow 0$

  trend,remainder$\leftarrow$ loess$(Y.bc)$

end

residue $\leftarrow Y.bc -$ trend $-$ seasonality

.

$\alpha, y \leftarrow$ holt.smoothingParameter (trend)

$\widehat{model}_t \leftarrow$ holt.model $_t$(trend, α, γ)

$\hat{t}_h \leftarrow \widehat{model}_t.forcast(trend, h)$

.

$\alpha, \gamma, k \leftarrow$ WinterHolt.smoothingParameter(seasonality)

$\widehat{model}_s \leftarrow$ WinterHolt.model$_s$(seasonality, α, γ, k)

$\hat{s}_h \leftarrow \widehat{model}_t.forcast(seasonality, h)$

.

[FeatureList] $\leftarrow ExtractFeatures(residue, window, FeatureList)$

model $\leftarrow$ **ModelSelection([FeatureList], residue)**

r˙h $\leftarrow$ model.forcast(residue, h)

.

$\overline{Y.bc} \leftarrow \hat{t}_h + \hat{s}_h + \hat{r}_h$

$\hat{Y} \leftarrow \mathrm{InvBoxcox}(\overline{Y.bc}, \lambda)$

Algorithm 2: Model Selection Algorithm for selecting the appropriate Machine Learning Method with lowest sMAPE

Function ModelSelection $([features], residue)$;
M $\leftarrow$ commitee of ML models
$\mathcal{M}_f \leftarrow [\dots]$
$sMAPE \leftarrow [\dots]$
for *model m in M* **do**
    model $\leftarrow TrainModel([features], residue)$
    $\mathcal{M}_f \cdot$ append (model)
    $y_{hat} \leftarrow$ model. forecast([features]$[-h:], h)$
    sMAPE. append (y_{hat} , residue$[-h:]$)
end
return $\mathcal{M}_f$[index$(SMAPE.\min())$]

Chapter 4

Experimental set up

In this section, we demonstrated our highly comparative proposed feature-based hybrid approach to time series out of sample point forecasting. We illustrated the method using monthly series from the M4 [42] and NN3 [43] competitions. These series are considered to have a variety of features such as seasonality, some exhibit a trend (exponential or linear), and some are trendless, just fluctuating around some level. We ran our experiments on both time series data. These allowed us to compare our result with published forecast results from these competitions, their benchmark techniques and also methods employed to create our hybrid methodology. Statistical measure such as MASE and sMAPE were used to evaluate the performance of this approach as explained in chapter 3. All performance comparison was based on 18 out of sample forecast points from these datasets using the iterative approach of multiple steps forecasting for the ML methods. Our reported model evaluation was limited to the statistical error metrics MASE, sMAPE and the Overall Weighted Average(OWA), as we only compared the error metrics and average rank of our proposed method with other statistical, machine learning methods and other researchers results. Therefore, statistical significance testing was not reported in this thesis, and the only statistical hypothesis testing that was carried out in this research was limited to model parameter selection with the hold out validation process [68]. Table 4.1 shows the range of observations of various group of series under consideration for this experiment.

Table 4.1: Dataset minimum and maximum number of observations over the 3-dataset used for this experiment

Dataset	**Count**	**Minimum Number of Observations**	**Maximum Number of Observations**
NN3 Reduced	11	133	144
NN3	111	68	144
M4 Monthly	48,000	42	2795
Total			

4.1 Parameter Determination

Little modification was done to the parameters of the model because of the large amount of data we experimented with, and the out of sample model validation as explained by [68] was carried out with the 11 NN3 reduced dataset due to the small number of time series data in the dataset, so that we can achieve an efficient search for parameter that helps improve the model in a controllable manner and also in a very easy fashion. Other ML parameters were kept at the default parameter of the implementation to reduce the computational complexity of the thesis. After several runs on our hybrid method, parameters such as smoothing parameters α, γ, k for the exponential smoothing algorithm, depth of trees t for the boosting algorithm and the size of the sliding window w were the three important parameters that influence the accuracy of our model.

The methods used for our hybrid models comprises of linear and non-linear functions, with the goal of our model to learn by solving an optimization problem by choosing to set of parameters that help minimize the error function, likewise producing models that have the overall generalization ability over a range of the diverse dataset. Due to the time dependency nature of our dataset, a hold-out out-of-sample method as explained by [68] was employed as a validation technique on NN3 reduced dataset to get the optimized parameter that provide an overall generalization. In the hold-out validation process, the last part of the training data was used for testing on several horizons ranging from short term to medium term and long-term horizon. We considered a suitable range of values for each parameter; for the sliding window w parameter, we depended on the forecast horizon h, so we tested for possibilities of $[h/2, h, h * 2, h * 3]$. The smoothing parameters α, γ, k are selected from a range of $[0.2, \ldots, 0.8]$ while the maximum depth of boosting trees max_depth from $[5, \ldots, 15]$ and the number of iterations of the algorithm $nrounds$ from $[1, \ldots, 250]$. Since the reduced NN3 dataset is just 11 series in numbers, we used a grid search approach to optimally select the parameters that achieved the lowest sMAPE and MASE before predicting the test dataset of the NN3 and M4 dataset. The final set of perimeters was $max\_depth = 5$, $nrounds = 100$ and sliding window $w = h * 2.5$.

4.2 Results

Results from Table 4.2 and Figure 4.1 shows the overall average and standard deviation of sMAPE and MASE for the different models compared on the total 11 NN3 datasets on long forecast horizon of 18 points.

Individual Methods	sMAPE	σ(sMAPE)	MASE	σ(MASE)
Our Hybrid Method	**14.17**	**12.09**	0.912	0.394
AutoArima	18.15	16.52	0.906	0.415
Holts	54.53	46.25	3.966	2.736
Winter	17.71	17.74	1.001	0.660
Gradient Boosting	36.11	39.06	1.983	1.104
Support Vector Regression	24.35	24.37	1.443	0.792
Combination Methods	**SMAPE**	**σ(SMAPE)**	**MASE**	**σ(MASE)**
Holts and Winter	21.22	16.74	1.495	0.992

Table 4.2: Forecast Performance and Standard Deviation on 18 point out-of-sample forecast on 11 NN3 Reduced Dataset

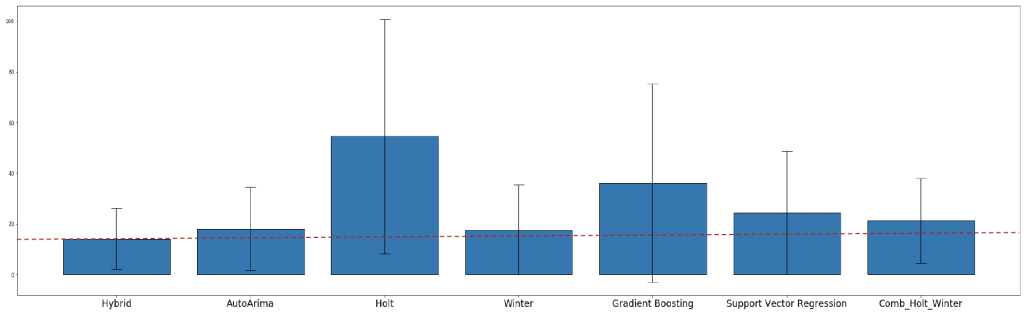

Figure 4.1: Forecast Performance and Standard Deviation of on the 11 NN3 Reduced Dataset using sMAPE performance metrics

The predictive performance of our proposed hybrid model was compared with other statistical methods, Machine Learning methods and also an automatic ARIMA model called AutoArima. AutoArima [69] automatically discovers the optimal order for an ARIMA model, a powerful and relevant tool for time series that requires appropriate parameter definition. AutoArima has the same time complexity compared to our hybrid approach, because it uses a grid search to try various sets of p and q (and also P and Q for seasonal model) parameter before selecting a model that minimizes the Akaike Information Criteria

(AIC) and Bayesian information criterion (BIC). The above result indicated that our proposed technique performed relatively better than AutoArima, Winter Holts and other ML methods on the NN3 reduced dataset. However, the machine learning algorithms did not perform so well when used alone for the forecasting task. Our feature based hybrid approach employed the combination of Holt and Winter method and from Table 4.2 above, our approach gave a higher performance measure than the linear combination of Holt and Winter method.

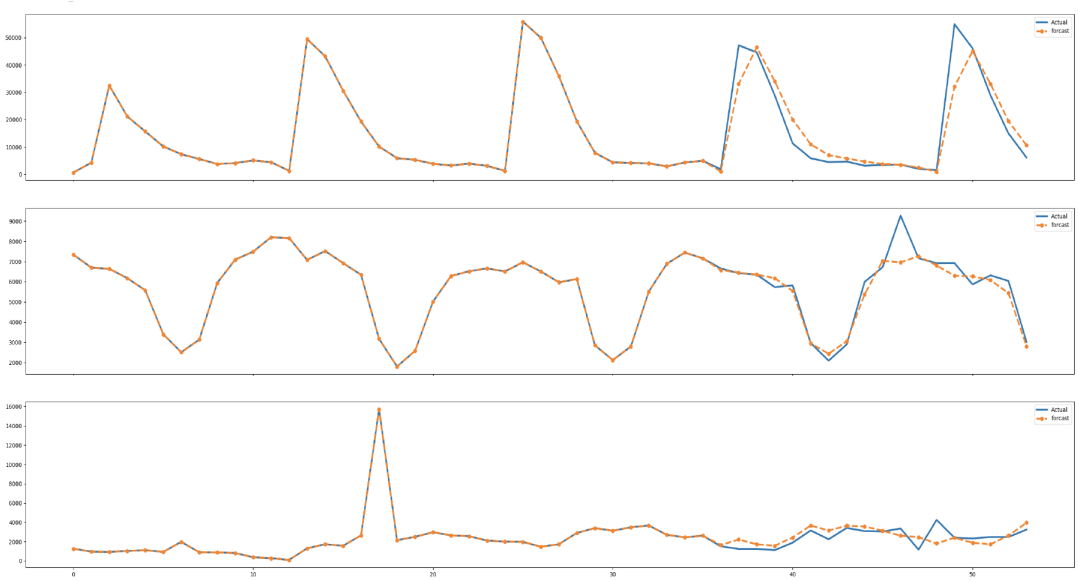

Figure 4.2: The forecast of our model on three of the NN3 reduced time series data. Shown is the trimmed training period, followed by the out of sample period (the last 18 points). NN3˙103 (top), NN3˙104 (middle) and NN3˙110 (bottom) with MASE of 0.60,0.37 and 0.94 respectively

This shows with results that a linear combination of these methods does not necessarily guarantee performance superiority, but a systematic combination as demonstrated by our methodology shows to produce more accurate out of sample forecast. We could conclude that our feature-based approach for ML methods and combination of statistical method, as implemented in our hybrid approach optimally increased the predictive performance of our model. To illustrate our presented ideas in a concrete manner, Figure 4.2 shows an example of the time series actual point, together with the out of sample point forecast produced from our proposed model.

Results from Figure 4.3 and Table 4.3 shows the mean and Standard deviation of the

Individual Methods	SMAPE	σ(SMAPE)	MASE	σ(MASE)
Our Hybrid Method	**16.28**	13.31	1.300	1.787
AutoArima	16.98	12.86	1.228	1.834
Holts	88.05	57.89	9.986	8.576
Winter	17.24	14.71	**1.172**	1.641
Gradient Boosting	19.10	15.67	1.355	1.848
Support Vector Regression	18.66	13.44	1.545	2.512
Combination Methods	**SMAPE**	**σ(SMAPE)**	**MASE**	**σ(MASE)**
Holts and Winter	51.17	48.02	3.642	3.047

Table 4.3: Forecast Performance and Standard Deviation on 18 point out-of-sample forecast on 111 Dataset

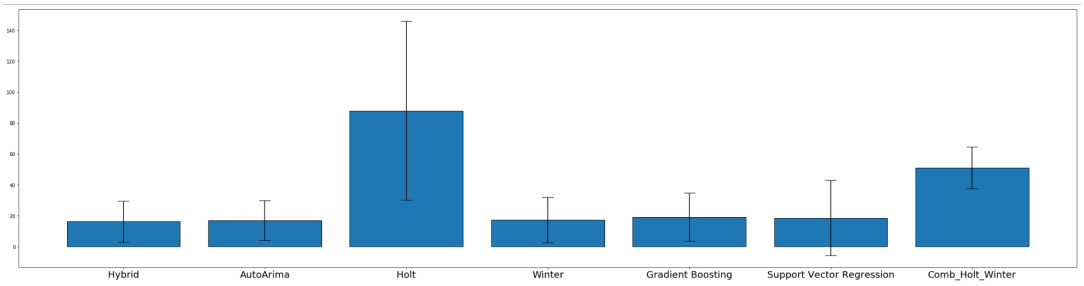

Figure 4.3: Forecast Performance and Standard Deviation of on the 100 NN3 Dataset using sMAPE performance metrics

performance metric of our proposed method against other comparing methods like traditional statistical models and Machine learning methods. From the results, it shows that our hybrid method and AutoArima have the best performance metrics and standard deviation, this implies that, all the sMAPE and MASE values of the individual series are close to the average sMAPE and MASE. Holt and combination of Holt and Winter gave erratic results with the worst performance. Perhaps this is because Holts method handles only trend component without consideration of other systematic component of the series, likewise, the combinatory approach of Holts and Winter does not have the flexibility as our proposed model to handle the complexity of complex real life time series data. The worst models overall on the NN3 dataset are Holts and linear combination as it consistently gave bad performance metrics.

Comparing our hybrid method with Machine Learning Models used as components of our proposed model, our model performed better that each on them because of our

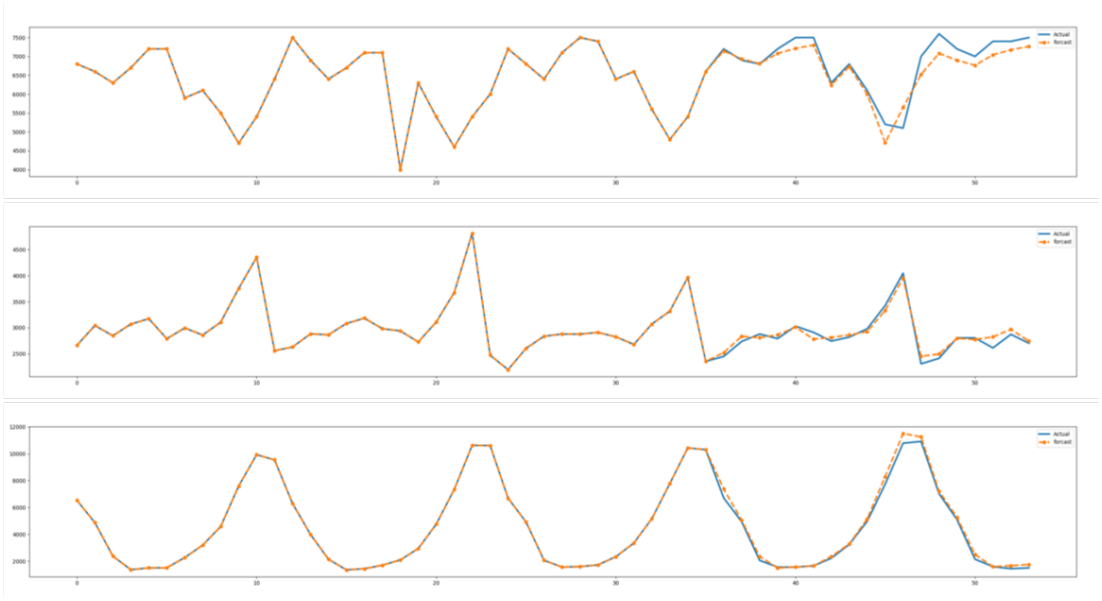

Figure 4.4: The forecast of our model on three of the NN3 100 time series data. Shown is the trimmed training period, followed by the out of sample period (the last 18 points). NN3˙058 (top), NN3˙068 (middle) and NN3˙059 (bottom) with sMAPE of 3.68, 2.83 and 6.01

approach of modelling which only considered the statistical features of de-seasonalised and detrended series, making the series stationary and perfect to be modelled by a ML model. This summarizes how effective the Machine Learning part of our proposed method is, with the help of statistical features extracted, our model was able to reduce the generalization error on these times series data thereby producing more accurate out of sample forecast.

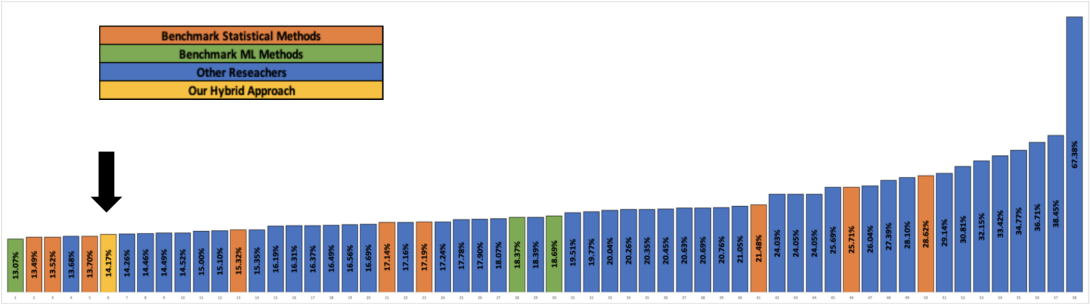

Figure 4.5: Histogram showing sMAPE of how our hybrid method compared with published result of the NN3 reduced dataset

We compared our result with the result submission from NN3 Competition [43],Figure 4.5 and Figure 4.7 shows our hybrid method relative performance with other statistical methods that entered the competition as benchmarks, novel methods from other researchers,

[1]Source: http://www.neural-forecasting-competition.com/NN3/results.htm

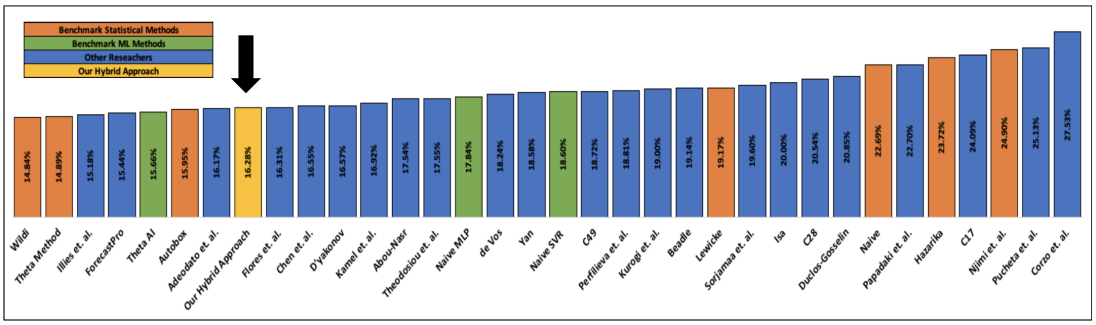

Figure 4.6: Histogram showing sMAPE of how our hybrid method compared with published result of the NN3 111 dataset. caption[1]

Machine Learning or Computational Intelligence (CI) methods. Most of the submissions from other research are novel methods that uses methods from computational intelligence methods and statistical methods or even both to automatically make 18 point out of sample time series forecast.

Results from Figure 4.5 and Figure 4.7 also shows that our hybrid model achieved among the top 6 models when tested on 11 NN3 reduced dataset, and among the top 8 models with the lowest sMAPE, when tested on 111 NN3 dataset. Among the top 8 models of the NN3 competition are commercially available softwares like CI Benchmark - Theta AI by Nikolopoulos [70], Stat. Benchmark - Autobox by Reily [71], Stat. Benchmark - ForecastPro by Stellwagen [72], Stat. Benchmark - Theta by Nikolopoulos [73], Stat. Contender Wildi [74] and and lastly Adeodato that applied Fourier analysis and Multilayer perceptron networks (MLP) for their predictive technique.

Our feature-based hybrid method performance is similar to the commercially available methods and well ranked as part of the top 10 approaches on all NN3 datasets, showing that our approach performance is not only comparable with individual state-of-the-art classical approaches but also with commercially available software used for forecasting.

To see more insight on how out of sample results from other models compared with results from our model, we tested the relative performance of our model with benchmark methods from the M4 competition. The M4 competition dataset contains monthly, quarterly, and annual series and has been used as wind-tunnel data for testing extrapolation methods. Figure 15 shows a visual representation of comparing our hybrid model out of point forecast with some the benchmarks methods of the M4 competition.

We computed the Overall Weighted Average (OWA) of the two accuracy measures; The

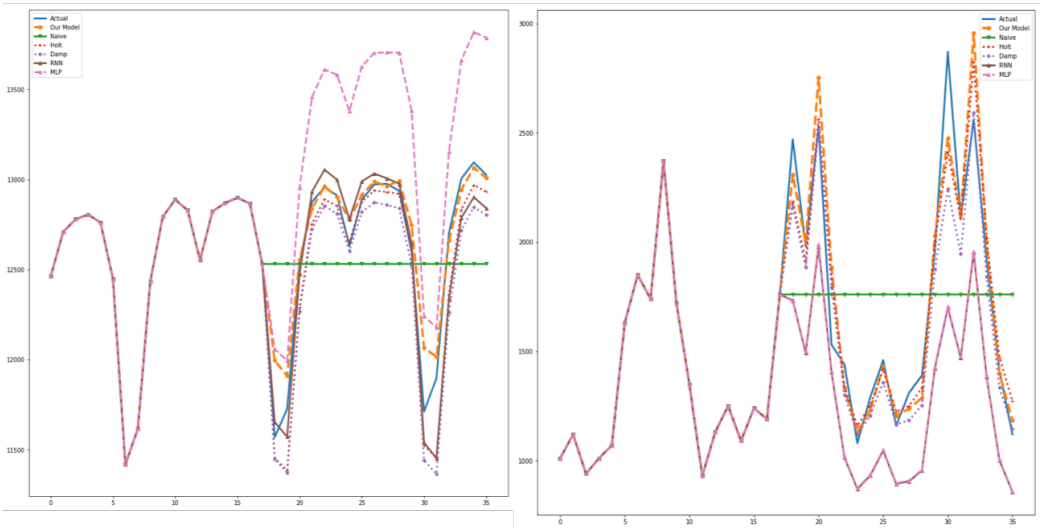

Figure 4.7: The 18 point out of sample forecast of our model and benchmark methods for one of the M4 time series. The training period is reduced to give more room for the out of sample forecast.M11- Macro (left) and M34- Macro(right) with sMAPE 0.78 and 6.49 respectively

Mean Absolute Scaled Error (MASE) and the symmetric Mean Absolute Percentage Error (sMAPE). The OWA is evaluated by first computing the relative MASE and sMAPE, which is dividing all the errors from all the methods by the error from the Naive method, and then averaging the relative MASE and sMAPE to obtain the OWA rank of the methods as shown in the below Table 4.4.

Method	sMAPE	Relative sMAPE	MASE	Relative MASE	OWA	Rank
AUTOARIMA	10.767	0.822	2.354	0.755	0.789	1
ETS- Benchmark	11.342	0.866	2.614	0.839	0.852	2
Our Hybrid Method	***10.944***	***0.836***	***2.900***	***0.930***	***0.883***	***3***
WINTER Method	12.034	0.919	2.935	0.942	0.930	4
HOLT- Benchmark	12.851	0.981	2.752	0.883	0.932	5
NAÏVE- Benchmark	13.096	1.000	3.117	1.000	1.000	6
Gradient Boosting	12.409	0.948	3.340	1.071	1.009	7
RNN- Benchmark	18.316	1.399	4.171	1.338	1.368	8
AVEARGE Holts and Winter	21.714	1.658	3.993	1.281	1.469	9
MLP- Benchmark	18.673	1.426	13.011	4.173	2.800	10
HOLT Linear Method	39.305	3.001	9.372	3.006	3.004	11
Support Vector Regression	25.727	1.964	12.618	4.047	3.006	12

Table 4.4: Table showing the computed OWA rank, and relative errors of our proposed model and other benchmark models.

Our proposed hybrid methods ranked as part of the top 3 forecasting methodswith relatively good performance after AutoArima and ETS as seen in Table 4.4. The AutoArima and the ETS model gave good performance accuracy by capturing all the dynamics in data as the residual component likewise our proposed model, which employs statistical prosperities and machine learning methods to give a good forecasting result. This shows that out of sample forecast predicted by our proposed method in this thesis is comparable with AutoArima and ETS methods with no much significant difference between their results.

Table 4.4 result clearly shows that Machine learning algorithms when used alone, are not very powerful to model time series data. Likewise, when compared with their traditional statistical counterparts, they do not give the best predictive performance. Perhaps, this is because time series data are mostly dominated by seasonality and randomness which have shown to be very difficult to be modelled by ML algorithms. They all had the highest OWA with the nave benchmark method performing better than all Machine Learning methods. Benchmark ML methods like Recurrent Neural Network (RNN) and Multilayer Perceptron (MLP) did not impress either as they could not outperform most statistical techniques and even the Nave approach. This was clearly stated as part of the major findings of the M4 competition [42]. Support Vector Machine and Gradient Boosting algorithm also performed woefully when used alone with high OWA, but combining them with statistical process improved their forecasting capability by a reasonable amount showing that when these same machine learning models are combined in a systematic manner with statistical process, they give less forecasting error as shown by our proposed methodology.

Lastly, we compared our proposed methodology with other combinatory techniques to determine the effectiveness of our proposed combinatory method. We examined our method with the metalearning approach for forecast combination developed by Lemke [75]. In their research, they implemented a newer approach of combination called ranking based combination method based on the zoomed ranking algorithm as explained in [76]. They explored their methodology on NN3 dataset and produced an in-sample forecast i.e. the test points used to evaluate the performance of their proposed methodology was used as part of the training set used for building the model.

Table 4.5 shows the performance result of our approach on the NN3 dataset, alongside methodology presented in [75], including individual methods employed for their ranking algorithm. Our proposed hybrid methodology gives a lower sMAPE measure than all the

Method	sMAPE
Decision Tree	18.00
Support Vector Machine	18.00
Zoomed ranking, best method	17.50
Neural Network	17.00
Our Hybrid Method	**16.26**
Zoomed ranking, combination	15.50

Table 4.5: Performance comparison between our hybrid method and different meta-learning techniques.

individual or Machine learning methods used for the ranking method. This result reinstates our claims that a combination of methods always performs better than an individual method in forecasting problem. However, the proposed combination zoomed ranking method performed relatively better than our proposed method in this research as it has a better sMAPE measure. This result is expected, considering the fact[2] stated in their experiment that the statistical test of the model performance was conducted using in-sample data period, while our proposed method performance was conducted using out-of-sample data period. Out-of-sample forecast performance is generally more trustworthy than evidence based on in-sample performance and also better reflect the information available to the forecaster in "real time". This result shows that our decomposition technique of combining time series forecasting is comparable to other combination method. Also, we could conclude with result presented that our methodology approach of extracting statistical components from the residual components of the time series data, improves the out of sample predictive capability of our model.

[2]The whole time series were used in the training set for building the models and a part of the training series were used to evaluate the model.

Chapter 5

Conclusion

In general, time series have shown to contain complex patterns that prove difficult to be model using simple statistic approaches and machine learning approaches alone. From research and results achieved, it shows that some kind of special treatment is mostly required based on the individual series to be able to analyze the different components of the series. Therefore, we proposed a new hybrid time series model for time series prediction that makes use of structural decomposition technique alongside Exponential Smoothing (ES) algorithm, feature extraction and Machine learning methods to perform long term horizon forecasting to a reasonable extent. The time series is decomposed into the seasonal, trend and residual systematic components. These components are then predicted separately and combined linearly to obtain the final predictions. We used the Holts method and Winter Holts variants of the Exponential Smoothing algorithm to predict the trend and seasonal components respectively. The prediction of the residual components, due to its irregular behavior was reinforced with time series feature extraction to get statistical information about the residual components, and then modeled with the sets of different Machine Learning algorithms such as Boosting Trees and SVR.

We tested our methodology on the 18 out of sample observation of the M4 and NN3 dataset and found that the error measure of our proposed feature based hybrid method is in the acceptable range and it outperforms most of the traditional statistical methods, Machine Learning methods and also the Artificial Neural Network (ANN) methods. Our work has also shown that our proposed method is comparable if not better than most individual statistical state-of-the-art and machine learning algorithms. It also points out techniques in which the machine learning methods can be combined with statistical methods to help improve the forecasting accuracy of time series data.

For this, we claim that our proposed methodology provides not only a higher generalization performance than these algorithms, but also show ways in which forecast combination could be better achieved without weights evaluations which most of the time are

very complex, difficult and bring more error into the system. It also shows with convincing results that feature-based approaches on the stationary component of the series achieved from STL decomposition helps reduce forecasting errors. Several researchers such as Neep Hazarika [77], Theodosiou and Swamy [78], among others have explored decomposition methodologies to perform multi-step ahead forecasting in the past with submission in the NN3 competition. However, our feature-based hybrid approach with lower sMAPE performance score of 16.26 outperformed these two with sMAPE performance score of 23.72 and 17.55 respectively. Perhaps, this performance could not be attributed to only the decomposition technique, but to the excellent capability of all the various methods such as, Exponential Smoothing methods, ML methods, and statistical feature extraction, that make up our hybrid methodology.

5.1 Discussion

This thesis investigated time series forecasting from a different point of view, using hybrid methods, and made contributions to answering research questions raised in the introductory chapter. The main general research questions are:

- ***How does the performance of our proposed hybrid methodology compare to the other individual state-of-the-art classical approaches?***

  Our hybrid method shows to have outperform all individual predictors on average throughout this research. This was evidently displayed from results in Table 4.2, Table 4.3, Table 4.4. However, caution is however necessary when applying combination techniques, as complex combinatory techniques have shown to sometimes infuse more errors into the forecasting model [44]. Our combinatory methods took advantage of advanced techniques such as STL decomposition and feature extraction to produce accurate point forecast. This supports our conclusion that individual method alone might not have an edge in empirical studies but could stand a chance through appropriate combination method. This conclusion was also confirmed by Makridakis in the just concluded M4 competition [42], "*The combination of statistical and/or ML methods will produce more accurate results than the best of the individual methods combined*".

- ***How effective are Machine Learning models in hybrid methodology for time series***

forecasting?

Results from chapter 4 shows the experimental results of machine learning methods on NN3 and M4 dataset, and from the result, we could see the forecasting strength of those machine learning method. On M4 dataset, even the Naive method performed better than the benchmark ML methods. These Machine Learning methods have shown not to be very strong predictors when used alone, mostly because of the various components such as seasonality and trend that are contained in time series data. However, our approach of decomposing a series into several components before extracting statistical features from the stationary residual components and modeling with ML techniques have resulted in consistent performance gains in this research. These shows how effective these ML techniques could be harnessed to improve the forecasting strength of a forecasting model.

- ***Does the decomposition of combinational approach compare to the other combination techniques?***

  This question was investigated using NN3 dataset in chapter 4, with our methodology compared with a meta-learning technique and results shown in Table 4.5. The Meta-learning technique exploited domain knowledge to improve forecasting, while our approach used automatic decomposition method to decompose the series into systematic components. These two methods achieved good results and both outperformed the individual methods employed in both approaches. From this result, we could conclude that our combinatory method is comparable to other advance combinatory techniques.

5.2 Limitations

The findings of this research have to be seen in the light of some limitations which are highlighted below.

- The short length of the observations in the NN3 and M4 dataset. However, the sample size might not be a problem to the statistical part of our model, but we needed more sample observation for the machine learning part of our hybrid model. This will have

enabled us to extract more statistical features that express the underlying properties of the series, thereby increasing the forecasting ability of our machine learning model.

- STL decomposition provides only additive decomposition but with Box-Cox transformation of the series, we were able to obtain multiplicative decomposition and then a back-transformation as explained in chapter 3

- Only positive observation series were observed, the reason being that the NN3 and M4 dataset comprises of only positive series. Transformation adjustment has to be done to forecast series with negative observation as a log transformation of negative data is undefined. This limitation will provide a major constraint on our methodology if we are to forecast intermittent series or time-series data with a combination of negative and positive observations.

- This research work was carried out on only monthly time series data and without consideration of any domain knowledge of the series or external factor influence that might affect the series at a particular point in time

5.3 Future Work

In future investigations, we will use our proposed hybrid on various types of time series data other than monthly such as daily, weekly, quarterly and yearly series. Our future work will also be centered around improving the efficiency and predictive performance of this algorithm in numerous ways such as

- improving the exploration of important features.

- Using an auto-adaptive parameter; one crucial consideration will be making the size of the box sliding window dynamic. This will help capture appropriately the dynamic statistical properties of the series being modeled.

- using domain features to enhance the efficiency of the proposed algorithm for solving domain specific problems such as climatology.

Bibliography

[1] C. Chatfield, *The analysis of time series: an introduction*. CRC press, 2004, vol. 59.

[2] J. Proakis and D. Manolakis, "Digital signal processing: principles, algorithms, and applications," *Digital signal processing: principles, algorithms, and applications by John G. Proakis, and Dimitris G. Manolakis. Upper Saddle River, NJ: Prentice Hall, 1996.*, vol. 1, 1996.

[3] R. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*, ser. Dover Phoenix Editions. Dover Publications, 2004. [Online]. Available: https://books.google.ca/books?id=XXFNW_QaJYgC

[4] B. M. Hales and P. J. Pronovost, "The checklist–a tool for error management and performance improvement." *Journal of critical care*, vol. 21, no. 3, pp. 231–5, Sep 2006.

[5] A. Haynes, T. Weiser, W. Berry, S. Lipsitz, A. Breizat, E. Dellinger, T. Herbosa, S. Joseph, P. Kibatala, M. Lapitan *et al.*, "A surgical safety checklist to reduce morbidity and mortality in a global population," *New England Journal of Medicine*, vol. 360, no. 5, pp. 491–499, 2009.

[6] T. C. Chamberlin, "The method of multiple working hypotheses," *Science*, vol. 148, no. 3671, pp. 754–759, 1965.

[7] J. S. Armstrong and K. C. Green, "Forecasting methods and principles: Evidence-based checklists," *Journal of Global Scholars of Marketing Science*, vol. 28, no. 2, pp. 103–159, 2018. [Online]. Available: https://doi.org/10.1080/21639159.2018.1441735

[8] M. Oliveira and L. Torgo, "Ensembles for time series forecasting," in *Proceedings of the Sixth Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Phung and H. Li, Eds., vol. 39. Nha Trang City, Vietnam: PMLR, 26–28 Nov 2015, pp. 360–370. [Online]. Available: http://proceedings.mlr.press/v39/oliveira14.html

[9] "Forecasting: Methods and applications, by spyros makridakis, steven c. wheelwright and rob j. hyndman. third edition. john wiley and sons, 1998, 642pp, isbn 0-471-53233-9. [uk pound]29.95."

[10] C. Lemke, "Combinations of time series forecasts : when and why are they beneficial?" 2010.

[11] G. P. Zhang, "A neural network ensemble method with jittered training data for time series forecasting," *Information Sciences*, vol. 177, no. 23, pp. 5329–5346, 2007.

[12] G. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[13] G. G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pp. 1–8, 2016.

[14] G. Box, G. Jenkins, and G. Reinsel, *Time series analysis: forecasting and control*. Wiley, 2011, vol. 734.

[15] R. H. McCuen, "Time series modelling of water resources and environmental systems: by k.w. hipel and a.i. mcleod. elsevier, amsterdam, 1994, hardcover, xxxvii + 1013 pp., dfl. 390., isbn 044489270-2," *Journal of Hydrology*, vol. 167, no. 1, pp. 399 – 400, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022169495900101

[16] R. T. Froyen, *Southern Economic Journal*, vol. 64, no. 1, pp. 337–339, 1997. [Online]. Available: http://www.jstor.org/stable/1061063

[17] G. Mahalakshmi, S. Sridevi, and S. Rajaram, "A survey on forecasting of time series data," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*. IEEE, 2016, pp. 1–8.

[18] G. Mbamalu and M. El-Hawary, "Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 343–348, 1993.

[19] R. Perrelli, "Introduction to arch & garch models," *University of Illinois Optional TA Handout*, pp. 1–7, 2001.

[20] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[21] J. F. Muth, "Optimal properties of exponentially weighted forecasts," *Journal of the American Statistical Association*, vol. 55, no. 290, pp. 299–306, 1960. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1960.10482064

[22] C. C. Pegels, "Exponential forecasting: some new variations," *Management Science*, pp. 311–315, 1969.

[23] I. Moghram and S. Rahman, "Analysis and evaluation of five short-term load forecasting techniques," *IEEE Transactions on power systems*, vol. 4, no. 4, pp. 1484–1491, 1989.

[24] E. S. Gardner Jr, "Forecasting the failure of component parts in computer systems: A case study," *International Journal of Forecasting*, vol. 9, no. 2, pp. 245–253, 1993.

[25] H. Grubb and A. Mason, "Long lead-time forecasting of uk air passengers by holt–winters methods with damped trend," *International Journal of Forecasting*, vol. 17, no. 1, pp. 71–82, 2001.

[26] T. Miller and M. Liberatore, "Seasonal exponential smoothing with damped trends: An application for production planning," *International Journal of Forecasting*, vol. 9, no. 4, pp. 509–515, 1993.

[27] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[28] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[29] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting: A hybrid lssvm and armax approach," *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 20–26, 2013.

[30] H. Fröhlich, O. Chapelle, and B. Schölkopf, "Feature selection for support vector machines using genetic algorithms," *International journal on artificial intelligence tools*, vol. 13, no. 04, pp. 791–800, 2004.

[31] R. Begg, *Neural Networks in Healthcare: Potential and Challenges: Potential and Challenges*. Igi Global, 2006.

[32] J. Faraway and C. Chatfield, "Time series forecasting with neural networks: a comparative study using the air line data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 47, no. 2, pp. 231–250, 1998.

[33] C. Hamzaçebi, "Improving artificial neural networks performance in seasonal time series forecasting," *Information Sciences*, vol. 178, no. 23, pp. 4550–4559, 2008.

[34] J. Kamruzzaman, R. A. Sarker, and R. K. Begg, "Modeling and prediction of foreign currency exchange markets," in *Artificial Neural Networks in Finance and Manufacturing*. IGI Global, 2006, pp. 139–151.

[35] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European journal of operational research*, vol. 160, no. 2, pp. 501–514, 2005.

[36] T. Taskaya-Temizel and M. C. Casey, "A comparative study of autoregressive neural network hybrids," *Neural Networks*, vol. 18, no. 5-6, pp. 781–789, 2005.

[37] M. Gan, H. Peng, and X.-p. Dong, "A hybrid algorithm to optimize rbf network architecture and parameters for nonlinear time series prediction," *Applied Mathematical Modelling*, vol. 36, no. 7, pp. 2911–2919, 2012.

[38] Q. Song, R. P. Leland, and B. S. Chissom, "A new fuzzy time-series model of fuzzy number observations," *Fuzzy Sets and Systems*, vol. 73, no. 3, pp. 341–348, 1995.

[39] E. Egrioglu, C. H. Aladag, and U. Yolcu, "Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks," *Expert Systems with Applications*, vol. 40, no. 3, pp. 854–857, 2013.

[40] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International journal of forecasting*, vol. 16, no. 4, pp. 451–476, 2000.

[41] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International journal of forecasting*, vol. 5, no. 4, pp. 559–583, 1989.

[42] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: Results, findings, conclusion and way forward," *International Journal of Forecasting*, vol. 34, no. 4, pp. 802–808, 2018.

[43] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction," *International Journal of forecasting*, vol. 27, no. 3, pp. 635–660, 2011.

[44] A. Timmermann, "Forecast combinations," *Handbook of economic forecasting*, vol. 1, pp. 135–196, 2006.

[45] F. X. Diebold and P. Pauly, "The use of prior information in forecast combination," *International Journal of Forecasting*, vol. 6, no. 4, pp. 503–508, 1990.

[46] G. Elliott, "Forecast combination with many forecasts," Mimeo, Department of Economics, University of California, San Diego, Tech. Rep., 2004.

[47] Y. Yang, "Combining forecasting procedures: some theoretical results," *Econometric Theory*, vol. 20, no. 1, pp. 176–222, 2004.

[48] C. Chatfield, *The analysis of time series: an introduction*. Chapman and Hall/CRC, 2003.

[49] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[50] V. M. Guerrero, "Time-series analysis supported by power transformations," *Journal of Forecasting*, vol. 12, no. 1, pp. 37–48, 1993.

[51] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[52] J. R. Halliday, D. G. Dorrell, and A. R. Wood, "An application of the fast fourier transform to the short-term prediction of sea wave behaviour," *Renewable Energy*, vol. 36, no. 6, pp. 1685–1692, 2011.

[53] S. Soltani, "On the use of the wavelet decomposition for time series prediction," *Neurocomputing*, vol. 48, no. 1-4, pp. 267–277, 2002.

[54] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: a seasonal-trend decomposition," *Journal of official statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[55] J. Brownlee, "Basic Feature Engineering With Time Series Data in Python," 2016. [Online]. Available: https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/

[56] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *Journal of machine learning research*, vol. 6, no. Sep, pp. 1621–1650, 2005.

[57] V. Cerqueira, L. Torgo, F. Pinto, and C. Soares, "Arbitrage of forecasting experts," *Machine Learning*, vol. 108, no. 6, pp. 913–944, 2019.

[58] C. Holt, "Forecasting trends and seasonal by exponentially weighted moving averages," *ONR Memorandum*, vol. 52, 1957.

[59] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management science*, vol. 6, no. 3, pp. 324–342, 1960.

[60] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[61] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[62] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[63] Vishal Morde, "XGBoost Algorithm: Long May She Reign! - Towards Data Science." [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[64] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[65] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[66] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[67] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, p. e0194889, 2018.

[68] V. Cerqueira, L. Torgo, J. Smailović, and I. Mozetič, "A comparative study of performance estimation methods for time series forecasting," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2017, pp. 529–538.

[69] "pmdarima: ARIMA estimators for Python pmdarima 1.2.0 documentation." [Online]. Available: https://www.alkaline-ml.com/pmdarima/index.html

[70] K. Nikolopoulos and V. Assimakopoulos, "Theta intelligent forecasting information system," *Industrial Management & Data Systems*, vol. 103, no. 9, pp. 711–726, 2003.

[71] D. Reilly, "The autobox system," *International Journal of Forecasting*, vol. 16, no. 4, pp. 531 – 533, 2000, the M3- Competition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207000000856

[72] R. L. Goodrich, "The Forecast Pro methodology," *International Journal of Forecasting*, vol. 16, no. 4, pp. 533–535, 2000. [Online]. Available: https://ideas.repec.org/a/eee/intfor/v16y2000i4p533-535.html

[73] V. Assimakopoulos and K. Nikolopoulos, "The theta model: a decomposition approach to forecasting," *International journal of forecasting*, vol. 16, no. 4, pp. 521–530, 2000.

[74] M. Wildi, "Nn3-forecasting competition: an adaptive robustified multi-step-ahead out-of-sample forecasting combination approach."

[75] C. Lemke and B. Gabrys, "Meta-learning for time series forecasting and forecast combination," *Neurocomputing*, vol. 73, no. 10-12, pp. 2006–2016, 2010.

[76] P. B. Brazdil, C. Soares, and J. P. Da Costa, "Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results," *Machine Learning*, vol. 50, no. 3, pp. 251–277, 2003.

[77] N. Hazarika, "Time series prediction using decomposition onto a system of random sequence basis functions and a temperature-dependent softmax combiner."

[78] M. Theodosiou and S. Murali, "A hybrid forecasting approach: structural decomposition, generalized regression neural networks and theta method," *NN3 Competition*, 2007.

[79] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

Appendix A

Programming Details

A.1 Libraries used

All the code is written in Python 3.7.2 and the following libraries are used in the project.

1. **Anaconda Python:** We used Anaconda version 4.3.30 in this project for this experiment. The Open source Anaconda distribution provides an easy way to perform data science and machine learning. It also helps manage library dependencies and environment and allows data analysis through Numpy and Panda. It also allows us to visualize our results using Matplotlib.

2. **Statsmodels:** We used the Statsmodels version 0.9.0 python module to provide the different statistical models used in our experiment and also for conducting statistical data exploration.

3. **Tsfresh:** To implement the feature extraction method, we used a python package called Time Series Feature extraction based on scalable hypothesis tests TSFRESH version 0.11.0 [79] to automatically extract features from time series that describe both basic and complex characteristics of the data. These statistical features of the residue components are used to train and build Machine learning methods that develop forecasting models used for predicting the next set of points.

4. **STL Decompose:** We used stldecompose version 0.0.3 to implement the STL algorithm.

5. **Scipy:** We used scipy version1.2.0 library for the Box-Cox transformation module.

A.2 Readme

In the interest of reproducibility, the methods and exert details of how the program needs to be run is publicly available in the GitHub repo: `https://bit.ly/2yyTBni`.