Real-time GDP Nowcasting in New Zealand

An Ensemble Machine Learning Approach

Author: JinJin Fan

Supervisor: Dr. Teo Susnjak

Co-supervisor: Dr. Andre Barczak

A Thesis Presented for the Degree of
Master of Philosophy
School of Natural and Computational Sciences
Massey University
New Zealand

ABSTRACT

Gross Domestic Product (GDP) measures the monetary value of all final goods and services that are produced in a region during a period of time. For most countries, GDP is released a limited number of times a year and often with a lag. Understanding the current economic situation, instead of figures quarters ago, is of vital importance for both policy and private entrepreneurs. It is crucial to create a live GDP predictor that could Nowcast current GDP growth rate in the period of government statistics release delay.

The Econometric approach for GDP Nowcasting has dominated the forecasting area for many years. However, most of the traditional econometric models could only incorporate a small handful of variables with a linear model structure, which could not meet the requirement of the "big data" era for a better model prediction ability with a large amount of unbalanced variables. With the improvement of computation ability and the increment of high frequency variables, data-driven approaches like Machine Learning Methods have been applied into Nowcasting area. It does not only show a stronger forecasting ability in handling large number of predictors but also present a superior robustness for non-linear data structure. In this research, an Ensemble Method constructed by several Machine Learning Methods have been generated to provide more timely available GDP figures in the period of government statistics release delay.

Having integrated an input dataset with data from multiple data sources such as public statistical websites, Reserve Bank of NZ and Stats NZ, our cooperators New Zealand Transport Agency (NZTA) and PayMark, this study is conducted by first applying different Machine Learning methods such as Lightgbm, Xgboost, Support Vector Machine, K-Nearest Neighbors, Ridge Regression, Lasso, Adaboost models. Then these algorithms are combined to generate an Ensemble Model with the assistance of an averaging method, which weights each model individually based on its historical prediction accuracy. The result of the final Ensemble Model is compared with the most commonly used benchmark ARIMA model and Random Walk model in terms of Mean Square Error (MSE) and Median Absolute Error(MAE) value. Statistical tests, such as Friedman Test and Wilcoxon Signed-Rank Test, are employed to check the significance of model superiority.

The results indicate that the Ensemble Model significantly outperforms individual

Machine Learning algorithm and Random Walk model in forecasting accuracy. When compared with the ARIMA model, it shows slightly better prediction ability with more fore-sights especially in a fluctuating environment.

Keywords: GDP, Nowcasting, Forecasting, Machine Learning

DEDICATION

This thesis is dedicated to my beloved mother and father, who have given their unconditional support to all of my tertiary endeavours. Studying abroad away from home is hard. Without their love and encouragement, this paper would not have been conceivable.

ACKNOWLEDGEMENTS

First, I would like to express my heartfelt gratitude to my supervisor, Dr.Teo Susnjak, who has lead me through the darkest period of my life and introduced me into such an interesting thesis topic, resulting in a very meaningful project. His expertise, patience, consistent guidance and confidence in me has helped to make this study successful.

I am deeply indebted to my dear friend, my respected teacher, Dr.Zhe Chen, for his encouragement and help all the way. As a researcher, his commitment and dedication to science has provided a good example for me to follow. As a friend, his firm belief and positive attitude to life has inspired me to overcome all the difficulty.

I wish to thank to the dearest staffs Dr.Andre Barczak, Tong Liu, Penny Lyall and Martha DeLong for the time and effort spent in helping me progress this thesis. Without their unselfish support and help, the thesis may not have been possible.

I would like to acknowledge my colleagues, Rahila Umer, Azadeh Izadi, Niloofar Aflaki, Gomathy Suganya, Sibghat Bazai Sibghat Ullah, Rongyao Hu, Jun Ren and many others. I benefited from our discussions and I am thankful for their ongoing support.

Most importantly, I am thankful for the support and constant encouragement from the Head of School of Natural and Computational Sciences, Prof Dianne Brunton. As a successful and respectable leader and friend, her earnest and positive attitude has not only influenced study, but my life!

I humbly extend my thanks to all concerned persons who cooperated with and supported me.

Contents

1 Introduction **1**

1.1 Background . . . 1
1.2 Problem Statement . . . 3
1.3 Scope and Study Objectives . . . 5
1.4 Research Questions . . . 5
1.5 Main Contribution . . . 6
1.6 Structure of the Research . . . 7

2 Econometric Approach for GDP Nowcasting **9**

2.1 Univariate Autoregressive Model . . . 10
2.1.1 ARIMA model . . . 10
2.1.2 Random Walk Model . . . 11
2.2 Vector Auto-Regressive Model . . . 12
2.3 Dynamic Factor Models . . . 13
2.4 Bridge Equation . . . 14
2.5 Mixed Data Sampling Regression . . . 15
2.6 Limitations of the Econometric Approach . . . 17

3 Machine Learning Approach for GDP Nowcasting **19**

3.1 Linear Methods . . . 20
3.1.1 Linear Regression . . . 20
3.1.2 Ridge Regression . . . 21
3.1.3 Lasso . . . 22
3.2 Support Vector Machine . . . 24
3.3 K-Nearest Neighbor Regression . . . 25
3.4 Tree Based Modeling . . . 27
3.4.1 Regression Tree . . . 27
3.4.2 Random Forest . . . 28
3.4.3 Adaboost . . . 30

3.4.4 Gradient Boosting . . . 31
3.4.5 Xgboost . . . 32
3.4.6 Lightgbm . . . 35
3.5 Model Averaging . . . 36

4 Methodology **39**
4.1 Tools . . . 39
4.1.1 The Jupyter Notebook . . . 39
4.1.2 Machine Learning Toolkit . . . 39
4.2 Workflow of Experiment . . . 40
4.3 Data . . . 42
4.3.1 Target Variable Selection . . . 43
4.3.2 Input Data Selection . . . 45
4.4 Data Interpolation . . . 46
4.5 Feature Engineering . . . 47
4.5.1 Feature Generating . . . 47
4.5.2 Feature Selection . . . 48
4.6 Data Shifting . . . 51
4.7 Forecast Evaluation Methodology . . . 52

5 Result Discussion **55**
5.1 Out-of-Sample One-step Nowcasting . . . 55
5.2 Expanding Window Validation . . . 62
5.3 Comparison with Random Walk and ARIMA models . . . 66

6 Conclusions **70**
6.1 Discussions . . . 70
6.2 Limitations and Delimitations . . . 71
6.3 Future Work . . . 72

List of Figures

1.1 Thesis Structure 8

3.1 Estimation Picture For The Lasso And Ridge Regression 23
3.2 K-Nearest Neighbors of A Data Point in A Two-Dimensional Feature Space 26
3.3 Decision Tree Growing Process 28
3.4 Comparison between Leaf Wise and Level Wise Tree Growth 34

4.1 Work Flow of Experiment Design 41
4.2 Plot of GDP Figures 43
4.3 Plot of Seasonally Adjusted GDP Figures 43
4.4 Dickey-Fuller Test 44
4.5 Correlation between Features 49
4.6 Distribution of All The Features 50
4.7 The Process of Data Shifting 52
4.8 The Process of Expanding Window 53

5.1 Training and Test Prediction Results Comparison 57
5.1 Training and Test Prediction Results Comparison 58
5.1 Training and Test Prediction Results Comparison 59
5.2 Prediction Results Comparison for Lightgbm, Xgboost, Gradient Boosting, SVM and the Ensemble Method from 2017-01-01 to 2018-12-31 64
5.3 Prediction Results Comparison for Decision Tree, Random Forest, Kernel Ridge, KNN and Adaboost Method from 2017-01-01 to 2018-12-31 65
5.4 Prediction Results of Ensemble Method, ARIMA and Random Walk from 2017-01-01 to 2018-12-31 Visualization 69

List of Tables

4.1 Simplified Procedure Of The Project 40
4.2 Variables For This Research 42
4.3 Brief Summary of Variable Statistics 42
4.4 Dickey-Fuller Test Results 45
4.5 Feature Summary 48

5.1 One-Step Prediction Evaluation: Interpolated Data Evaluation 61
5.2 One-Step Prediction Evaluation: Real Data Evaluation 61
5.3 MSE values of All the Algorithms In The Expanding Window 63
5.4 Algorithm Ranking in the Expanding Window 63
5.5 MSE value for the Real Data Evaluation in The Expanding Window 66
5.6 Algorithm Ranking for the Real Data Evaluation in The Expanding Window 67
5.7 Wilcoxon Signed-Rank Test 67
5.8 Prediction Results Evaluation from 2017-01-01 to 2018-12-31 68

Chapter 1

Introduction

1.1 Background

The problem that many companies, who rely on the national economical situation to make development strategy are confronted with, is that some financial variables are released with limited frequency and often with a lag. In most cases, the statistical information published by the government is from a few months ago on average, which is an inconvenience for decision makers such as the government, businesses, media and other institutions to understand how the economy performs currently.

Among all the financial indicators, one of the most important variables is Gross Domestic Product, namely GDP, which measures the market value of all final goods and services that are produced in a region during a period of time. The region is most commonly a country, and the period of time is defined as a quarter since GDP figures in most countries are released once a quarter.

The equation to calculate GDP is: $GDP = I + C + G + (X - M)$, where I stands for Gross Investment, C refers to Private Consumption, G represents Government Investment and $X - M$ is the Exports Income, removing the Imports Cost. Calculating GDP requires quantifying the evaluation of productions from every industry. There is usually a two to three months delay for the government to release national and industry figures. In addition to the delay, GDP values are frequently revised afterwards.

In New Zealand, the official estimates of GDP are released quarterly and with approximately a three-month lag. After the end of a calendar quarter, the estimation of the completed quarter of national GDP is released on around the 20th of the beginning month of the next quarter, which refers to March, June, September and December. In the case of Regional GDP, these figures are made available only once a year.

As Samuelson described, 'GDP has the ability to provide a good measure of the state of an economy like a weather satellite examines the entire continents' weather[1]'. GDP is

the universal standard measurement for a nation's economic growth and for the comparison of economies between countries. The information GDP contains, the value of total production, expenditure or income of an economy, adequately signals the performance and the state of a society. High GDP values reflect a healthy economy, within which the unemployment rate is low, wages are on the rise as well as the labour and production demands. Investors are correspondingly more willing to invest, and thus in turn leads to a higher level of prosperity. Studying GDP value could help policy makers and private decision makers better understand whether an economy is contracting or expanding and respond if necessary to avoid recession or inflation.

Given the availability of high frequency data and recent technological advances, it seems frustrating that when GDP figures are released we still do not know how the economy is performing right now but only how it was several months ago. To fill in this gap, a real-time and more up-to-date indicator of economic activity that could be updated in time and more frequently during the official publications delay, is in great demand. Nowcasting models for GDP are the effective answers to predict current economical growth, therefore, mitigate these uncertainties caused by the official publications delay.

"Nowcasting" is a contraction of the terms Now and Forecasting, implying that it attempts to "forecast" or estimate the present. Giannone, Reichlin, and Small define it as: 'The prediction of the present, the very near future and the very recent past'[2]. It differs from Forecasting in the sense that it focuses only on the next point in time to forecast. According to Bańbura, Giannone, Modugno, *et al.* the current quarter is the only one in which models produce significant gains relative to naive constant growth models for the GDP, thus motivating the focus on Nowcasting[3]. Moreover, Nowcasting deals with problems not usually dealt with by models attempting to generate long term forecasts.

Traditionally, the area of GDP Nowcasting is dominated by econometric methods like time-series regression models. Prompted by advances in computing power, Machine Learning methods with their remarkable performances in dealing with big data have recently been proposed as powerful alternatives with higher forecasting accuracy and stronger model generalization[4].

Machine Learning generally refers to the changes in system that perform tasks related with Artificial Intelligence (AI). The tasks include recognition, diagnosis, planning, and predictions amongst others[5]. Machine Learning utilizes Artificial Intelligence to enlist rules and describe patterns that the analyst can apply to new data. Once a model performs well on previously seen data, the analyst can feed in new data and the model can be used to predict and understand aspects of newly observed data[6]. All the processes involved are automated to deliver more accurate estimations quickly when compared to traditional econometric methods[7].

Machine Learning algorithms can be categorized as Supervised or Unsupervised Learning, based on the type of input data. Supervised learning examines records that have a known outcome. For example, supervised learning is used to study the academic behaviour of students with the intent to link student behavioural patterns to academic history and other recorded information. Unsupervised Learning is used in a situation where the patterns are unknown. Unsupervised learning is utilized first to study the patterns and look for previously hidden patterns, and to understand, classify and code the objects of study before applying theories [7]. For the GDP Nowcasting problem, the GDP figure is set as the target variable, and the approximation result is used to compare with government released values. This project focuses on supervised learning, particularly predictive analysis, where Machine Learning is utilized to predict future outcomes[8].

1.2 Problem Statement

The idea of Nowcasting is that by drawing on a large set of high-frequency sources (e.g. unemployment rate, price index, vehicle numbers on the motorway, etc.), signals about current GDP can be extracted before the associated official GDP figures are published. The availability of abundant data in macroeconomics provides an opportunity for researchers to include more relevant data into models to improve forecasting[9]. Adding a large and timely data set which are correlated with the target variable will maximize the information and improve model forecasting accuracy while on the other hand, it introduces a trade-off between bias and variance of the selected model. Incorporating more data into a fitted model might reduce bias but at the same time, make the model less generalized i.e. increasing the variance. The explosion in the amount of data in the digital age have made guaranteed the availability of large scale of high frequency predictors while including high frequency data for Forecasting or Nowcasting would introduce three main problems: the mixed frequency problem, parameters proliferation problem and the unsteady model selection problem.

- **The Mixed Frequency Problem.** In macroeconomic research, different publication periods of macroeconomic variables and non-matched data sources makes mixed frequency data very common. Many data are published on a quarterly frequency, such as GDP and CPI (Consumer Price Index), while other indicators are released on a monthly or weekly measurement, like vehicle numbers on the state high ways. Data from corporate sources, such as transactional data from Paymark, is available daily. This situation creates a challenge for researchers that, on a certain time point, differences in the availability of data make the data set unbalanced. Realigning the data into the same frequency is the solution for the mixed frequency data problem.

Traditionally, aggregation is one common way to deal with mixed frequency data. For a stock variable, the standard approach is to average the observations or to take the latest available observation of high frequency data to match up low frequency. For a flow variable, the normal way is to sum up the latest available observations in the current period of interest. Nevertheless, the reduction of the observation frequency will reduce the degree of freedom for a fitted model. Alternatively, we employ interpolation to the low frequency data to match data frequencies, which keep the basic trend of the original input, at the same time, will create a larger dataset and make the model more flexible in its handling of the high dimensionality problem.

- **Parameters Proliferation Problem.** Parameters proliferation ("curse of dimensionality", or, fat regression) refers to the case where N is greater than T, where N is the number of regressors, and T is the number of observations on the time series. In this situation, the number of regressors is so large that there is not enough data available for the computation in terms of degrees of freedom. Many solutions have been developed to reduce the dimension such as Bridge models, MIDAS with weighting schema [10], Factor MIDAS[11], PCA [12], Ridge Regression, and Lasso regression[13]. We incorporate penalized regression approaches, such as Ridge Regression and the Lasso (Least Absolute Shrinkage and Selection Operator) method, Bagging and Boosting methods to reduce the dimension of the parameter space. De Mol, Giannone, and Reichlin(2008) show that for the forecasting of macroeconomic panel data, a Bayesian shrinkage method is a valid alternative to principal components[13]. Bai and Ng(2009) suggest a way of using boosting to select variables in a factor model setting, and show that some forms of boosting outperform the standard factor-augmented forecasts[14]. Buchen and Wohlrabe (2011) show that boosting is a serious competitor for forecasting US industrial production[15].

- **The Unsteady Model Selection Problem.** Timmermann(2006) and Stock and Watson(2009) proposed a list of forecast combination methods. When there are 'n' possible input variables, there are '2^n' possible models in the model space. Selecting the best algorithms from the entire model space is very demanding. Furthermore, the parameters might not be constant with time. These issues cause uncertainty and instability of the model. The combination model integrates the advantage of all possible forecasting models rather than a single specific model, which helps to overcome the the problem of model uncertainty[18]. A model could not be regarded robust without solving the problem of the uncertainty and variability. More stable results could be generated by forecast combinations instead of single models, which perform better in dealing with model instability and structural breaks. In this paper,

the predictions are combined using a weighting method called system averaging, where the model prediction with least historical error receives the largest weight in the final future prediction.

1.3 Scope and Study Objectives

Primarily, Nowcasting the economic growth research is to obtain a better understanding of how GDP growth rate develops currently when official figures have not been released yet. A large data set of high frequency macroeconomic variables as well as other leading proxy variables which contain effective information to capture the economic environment change are employed to make the approximation. In this process, the model interpretation or the causal relationship between GDP quarterly change and indicating variables are less important than the forecasting accuracy. Therefore, generating a model to minimize the error between quarterly GDP change in New Zealand and the predicted results is the prior motivation and the scope of this study.

The main objective of this research is to create a live GDP predictor that could nowcast current GDP growth rate in New Zealand based on a group of macroeconomic variables which are more timely available than public GDP figures. This will be implemented by first applying different Machine Learning Methods such as K-Nearest Neighbors algorithms, Ridge Regression, Lasso, Adaboost models etc. and then combining these algorithms with the assistance of a system of averaging model, which weights each model individually based on its historical prediction accuracy. The results (MSE and MAE value) of our final model are compared with the most commonly used model ARIMA and Random Walk Model in previous work as well as each single Machine Learning algorithm.

1.4 Research Questions

- Is it possible to predict real-time GDP increment on the dataset gathered from our cooperators and public websites? If so, how to solve the mixed frequency problem caused by different data source? On which frequency basis (daily, monthly or quarterly) would the prediction be conducted?

- Is it possible to generate an accurate model to predict a real-time GDP increment? If so, which input variables or features should be included in the model? How could we select the algorithms? How could we tune the parameters? Which evaluation matrix shall be used?

- Is is possible to build a robust model to predict real-time GDP increment? If so,

how could we solve the problem of uncertainty and variability? How could we prove the model's robustness?

1.5 Main Contribution

The model we have built is a world-first real-time GDP forecaster, which uses big data technologies and data-driven techniques to form estimates of economic activity in New Zealand. The main contribution of this project includes three aspects:

- **Integrate a large set of input dataset from multiple data sources.** Given the delay in the publication of macroeconomic data and the unbalanced period of different data sources, many previous researches just employ proxy variables such as city night illumination [19][20], traffic fatalities [21], liquidity shocks resonating in stock and housing markets[22], etc. to measure the economical situation. Although, the proxy indicators could present some information on the current economic situation, they could not capture the underlying change in time. Our project has been honorably supported by a range of key New Zealand organizations from a variety of industry sectors. Our partner Paymark, from whom data is collected and fed into the Machine Learning Models on a daily basis, has made the final model possible. Our work also incorporates publicly available data from Ministry of Business, Innovation & Employment, New Zealand Immigration, NZ Transport Agency, Reserve Bank of NZ and Stats NZ. In summary, public macroeconomic variables such as unemployment rate, the consumers price index etc., proxy indicators such as traffic flows on the motorway as well as daily transactional data from our partners have been included in our real-time GDP forecaster.

- **Build a data-driven model with all kinds of machine learning regressors for GDP prediction in New Zealand.** Limited by the unvanquishable computational difficulties and the small size of macroeconomical indicators, the time series forecasting area for a long time has been dominated by linear statistical methods such as ARIMA models. Until the late 1970s and early 1980s, it became gradually clear that the linear models are not adaptive in many real world applications[23]. In the recent two decades, the most eye-catching and rapidly developing field is Machine Learning. Letting the machine find patterns in the input dataset and make predictions by those patterns has been proved to be more robust and competitive than traditional statistic models. An obvious advantage of the data-driven approaches like Machine Learning is that they can easily deal with a great number and diversity of data. They are also more adaptive to the non-linear data structure. Werbos demonstated that Artificial Neural Networks (ANNs) exceeded classic statistical methods

in time series forecasting and Lapedes and Farber concluded that ANNs perform well for modeling nonlinear time series. Other models such as Decision Trees, Support Vector Machines, K-Nearest Neighbors have also outperformed in a number of forecasting competitions[26].

The forecasting performance obtained from different data-rich methods differs widely [27]. Given the integrated input dataset, with the assistant of high frequency variables from different data sources, we have tested eleven Machine Learning Algorithms and recorded their prediction accuracy while most of New Zealand institutions are using classic statistic models such as Linear Regression and ARIMA model.

- **Create an Ensemble Model which combines all the Machine Learning Models together with a system of averaging method.** There is no 'perfect' algorithm adaptive to different problem domains. Underlying economic conditions change with time, which cause complexity in model selection. Besides, there are numerous parameters to tune for a sophisticated algorithm. All the variation contributes to instability of the model. Empirical evidence has shown that the accuracy and the reliability of a forecasting model could be improved significantly by combining multiple individual models together. The Ensemble Model we generate weights the prediction results from the nine out of the eleven Machine Learning Models and summarizes into a single prediction, which greatly improved the robustness of the prediction accuracy.

1.6 Structure of the Research

The structure of this paper is organized as Figure 1.1. In the area of GDP Nowcasting, most researches are conducted by by econometric models. Chapter 2 provides a brief introduction of some popular GDP nowcasting econometric approaches and their strengths and weaknesses. Chapter 3 presents the advantages of Machine Learning methods in managing the challenges raised by big data and improving nowcasting accuracy. Meanwhile, Chapter 3 lists the theory of Machine Learning Regression Methods applied in this thesis as well as their application in Forecasting. Chapter 4 is the methodology which introduces the process we collect data, conduct data pre-processing, apply feature engineering and build the models. The prediction results and the comparison with benchmark models are discussed in Chapter 5. Our conclusions and suggestions for future study are given in Chapter 6.

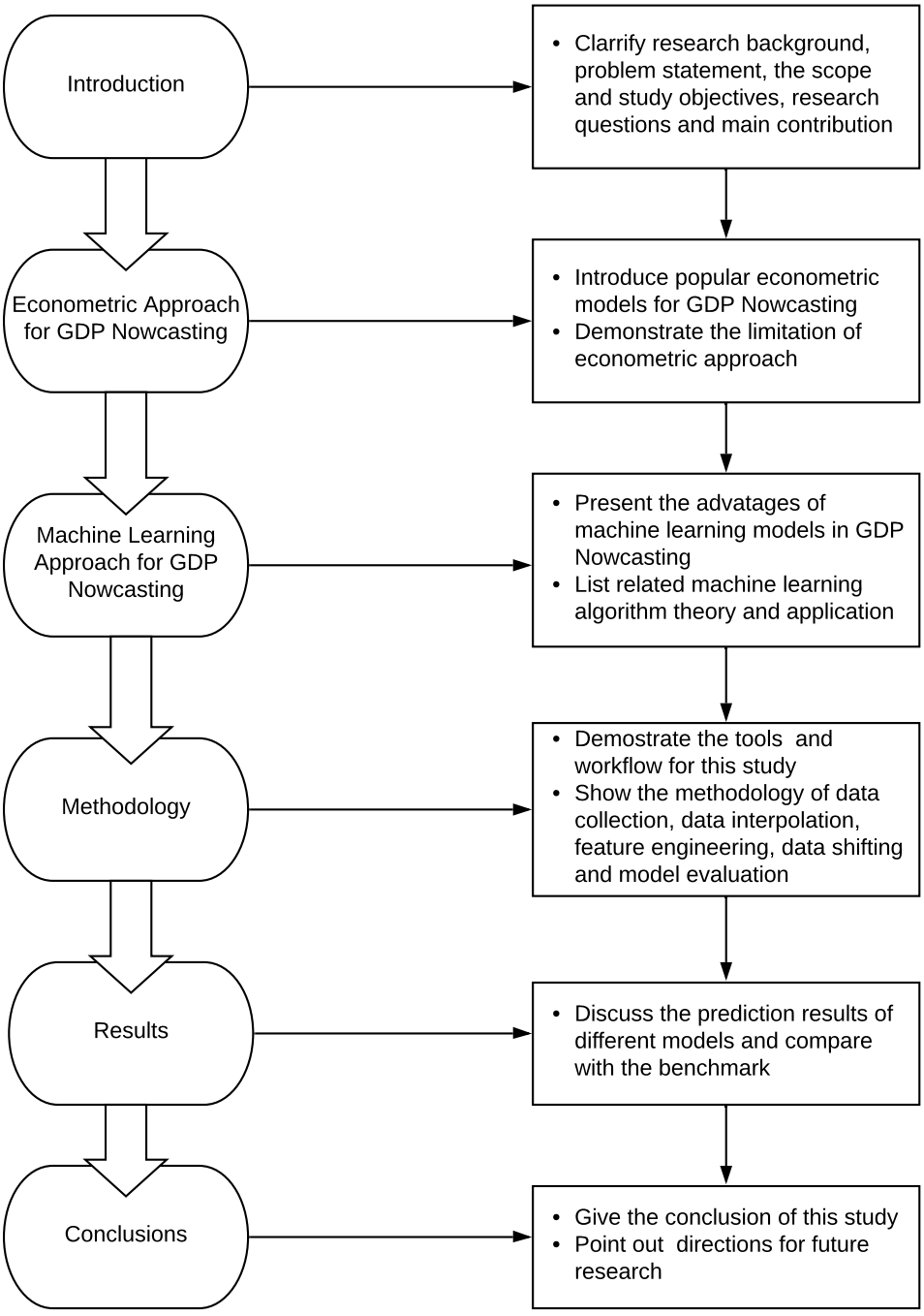


Figure 1.1: Thesis Structure

Chapter 2

Econometric Approach for GDP Nowcasting

Nowcasting combined by the terms "Now" and "Forecasting", has recently become popular in economics prediction, meeting the increasing demand for timely short term analysis and forecasting of the economic situation. Key measurements of macroeconomics data like GDP are normally released less frequently and with a time lag. Also they are then subject to subsequent revisions. This situation therefore generates the need to use available, timely and reliable information to create short term approximation for the key variables of interest[28].

There is a vast of studies on Nowcasting in GDP change i.e economic growth by now. Many nuances and increments exists but broadly speaking, two main directions arise, those who use it descriptively to support theory and those who "let the data speak". The former one is the Econometric Approach and the latter one is the Machine Learning Approach. The Econometric Approach has achieved remarkable forecasting performance in GDP Nowcasting especially for small and balanced datasets. Besides, the econometric practices have been primarily focused on exploring the causality between the indicators and the target variable in order to provide model interpretation for policy makers. With the increasing availability of related data from multiple source and a higher requirement of the forecasting accuracy rather than the interpretation of a forecasting model, traditional econometric approach shows some limitations.

In this Chapter, several popular Econometric Approaches for GDP Nowcasting are introduced. Generally speaking, Nowcasting with an Econometric Approach relies on either a Univariate Autoregressive Approach or Regression Based Methods like Bridge Equation, the Mixed Data Sampling (MIDAS) to handle mixed frequency and Vector Autoregressive Models(VAR) or Factor Models to handle other data irregularities. A

summary of the limitations in Econometric Approaches is presented at the end.

2.1 Univariate Autoregressive Model

In the case of GDP forecasting, a simple and powerful method is the Univariate Approach like AR, ARIMA model which approximates GDP growth only based on its autoregressive terms.

2.1.1 ARIMA model

The ARIMA model is the basic and most general ideology applied for analyzing and forecasting time series data[29]. It provides a simple but powerful method for capturing auto-correlation in the series by modelling it directly. ARIMA stands for AutoRegressive Integrated Moving Average, which is a descriptive acronym including the key aspects of the model itself. It is a series which needs to be differentiated in order to be made stationary is an "integrated" (I) series. Lags of the stationarized series are called "autoregressive" that refers to (AR) terms while lags of the forecast errors are called "moving average" which refers to (MA) terms.

An ARIMA(p,d,q) model generally contains three parameters, where:

- **p** is the number of autoregressive terms
- **d** is the number of nonseasonal differences needed for stationarity
- **q** is the number of lagged forecast errors in the prediction equation

The forcasting equation is structured as follows. First, transform the time series to be stationary by the differencing process. Let y denote the dth difference of time series data Y, to be specific:

If d=0, $y_t = Y_t$

If d=1, $y_t = Y_t - Y_{t-1}$

If d=2, $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$

The second difference of Y when d=2 is the difference of the first difference, which is the discrete analog of a second derivative, the rest can be done in the same manner.

In terms of y, the general forecasting equation is:

$$\hat{y_t} = \mu + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \theta_1 e_{t-1} - ... + \theta_q e_{t-q} \tag{2.1}$$

Here, p denotes the order of the autoregressive part, which allows one to incorporate the effect of past values into consideration. q represents the order of the moving average

part, constructing the error of ARIMA model as a linear combination of the error values observed at previous time points in the past.

A value of 0 is acceptable for each parameter which indicates to not include that element of the model. In this case, ARIMA model can be configured to perform the function of an ARIMA model or even a simple AR, I or MA model.

Introduced by Box and Jenkins [30], ARIMA has been widely used in the forecasting area.It is an adaptive data-oriented approach which captures the structure of the time series itself [31].

In 2001, Vincent Cho has investigated three methods exponential smoothing, univariate ARIMA and adjusted ARIMA to forecast the number of tourists from different countries to Hong Kong. The results showed that exponential smoothing is the worst considering the prediction accuracy, and univariate ARIMA as well as adjusted ARIMA model is more suitable for fluctuating time series[32].

Goh and Law have compared seasonal ARIMA model and monthly ARIMA model with the other eight time series models in forecasting tourism demand for Hong Kong. SARIMA (seasonal ARIMA model) is proved to gain the highest accuracy over all the methods while the simple ARIMA performs above the average of all forecasting models[33].

Kandananond has tested the forecasting ability of ANN, ARIMA as well as multiple linear regression(MLR) to predict electricity demand of Thailand. Statistical test showed that there was no significant difference among the accuracy of the three models while ARIMA and MLR model might beat ANN in the simplicity of model structure[34].

Twanabasu and Bremdal have applied ARIMA, ANN and SVM to study load forecasting in a smart grid oriented building. The ANN model outperformed ARIMA and SVM with a small mean absolute percentage error (MAPE) followed by ARIMA. However, the ARIMA model was selected because of its transparency[35].

A combined model of ARIMA and ANN was generated by Zhuang, Chen, Shi, *et al.* on predicting cooling load in 2015. The analysis demonstrated that the ensemble model is superior to simple time series models[36].

Despite the wide application of ARIMA model in Forecasting, as mentioned in the overview above, in an ARIMA model, the differenced future value of the target variable is constructed with a linear combination of past values and past errors. Thus, any significant nonlinear data trend would set limitation to ARIMA.

2.1.2 Random Walk Model

A pure Auto Regressive (AR) model is the one assumes that future data only depends on its historical trend,i.e. Y_t depends only on its own lags. AR(1) is also called random walk model,which is the simplest possible model for a time series data prediction. In each time

period, the next step is only decided by the previous data point and takes an independent random step away[37]. The equation is written below:

$$\hat{Y_t} = \varepsilon_t + Y_{t-1} \tag{2.2}$$

Where Y_{t-1} and Y_t are the observations of the time series and ε_t is a white noise with zero mean and a constant variance.

It can be seen from the equations that random walk model is quite straightforward and easy to implement. All the information about future trend is gained from the currently available data[38]. However, random walk model is commonly believed as a simple but effective model in financial area like foreign exchange rate forecasting[39][40][41]. Previous studies discovered that many elegant and sophisticated models in the linear domain do not outperform the naive Random Walk model[42].

To sum up, due to its simple structure and effective forecasting ability, Univariate Autoregressive Model has gained popularity in time series data forecasting. Meanwhile limited by its linear characteristic, in recent researches, ARIMA has more often been applied to compare with other models as a benchmark or to construct a hybrid model capturing the linear domain. In this thesis, ARIMA and Random Walk Model are selected as the benchmark to compare with other Machine Learning algorithms in Forecasting accuracy of GDP Nowcasting. However, there is an apparent shortcoming of the simple Univariate Approach. Mainly because it predicts the economic growth only relying on its historic data vintage which has not taken the possibility of modelling the complex economic system based on leading indicators into consideration. The forecasting ability of Univariate models like ARIMA could be effective during the stable period of economic development while it lacks of the foresight for a hidden turning point like the Great Depression.

2.2 Vector Auto-Regressive Model

A natural extension of the Univariate Auto-Regressive Model is the Vector Autoregression model (VAR), which was proven to have superior forecast ability compared to Univariate Time Series Models and elaborates theory-based simultaneous equations models[43]. Introduced by Sims in 1980[44], Vector Models can create structural equations with other influential features, a limited number of indicators is included in the model. The general form of the VAR(p) model with deterministic term and exogenous variables can be expressed as

$$Y_t = \Pi_1 Y_{t-1} + \Pi_2 Y_{t-2} + ... + \Pi_p Y_{t-p} + \Phi D_t + G X_t + \varepsilon_t \tag{2.3}$$

where D_t denotes for an $(l \times 1)$ matrix of other deterministic terms such as linear time trend or seasonal dummy variables and X_t stands for an $(m \times 1)$ matrix of stochastic exogenous components, Φ and G are parameter matrices.

Mittnik and Zadrozny in 2005 have estimated VAR(2) models of quarterly GDP and up to three monthly indicators variables which incoporate all significant correlated data for forecasting German real GDP at monthly intervals[45]. Caraiani *et al.* in 2010 have found the Bayesian VAR framework outperforms the OLS and the unrestricted VAR in forecasting the dynamics of GDP quarterly change for the Romanian economy[46]. Schorfheide and Song in 2015 have evaluated forecasts from the mixed-frequency VAR and compare the results with the standard quarterly frequency VAR and the MIDAS regression[47].

Besides its favorable ability in data description and forecasting, VAR model is also applied for structural inference where certain assumptions about the causal structure of the data under investigation are imposed. VAR model is therefore useful for impulse response analysis between variables, but not necessarily predicting GDP from a large vector of indicators. Thus, the choice of variables of interest influences the forecasting results[48]. The advantages of VAR models with a set of parameters to be estimated are at a greater cost of the parameters uncertainty[49].

2.3 Dynamic Factor Models

Macroeconomic data is often "wider" than it is "length", meaning that a lot of different variables exists but each containing relatively few observations. The econometric solution Dynamic Factor Models, which solve some of these problems, as they allow for many indicators to be included in the prediction.

Generally, factor models reduce dimension in a large data set by summarize the information available into a small number of factors. Each time series variable is represented as the sum of two components: the common component and the idiosyncratic component. The common component constructed with a linear combination of the common factors could explain the main part of the variance of the time series. The idiosyncratic component contains the remaining variable specific information[50]. The set of variables X_t could be written as an approximate linear dynamic factor representation with common factors f_t as:

$$X_{it} = \lambda_i(L) f_t + \varepsilon_{it} \tag{2.4}$$

where ε_{it} denotes the idiosyncratic disturbance with limited cross-sectional and temporal dependence and $\lambda_i(L)$ is the lag polynomial term in non-negative power of L which represents the vector of dynamic factor loading.

Empirical literature proves that dynamic factor forecasts are usually superior to simple

time-series techniques such as Univariate models and VAR. Artis, Banerjee, and Marcellino in 2005 have modelled a large macroeconomic data for the UK with a dynamic factor and demonstrated that the application of a dynamic factor model leads to considerable improvements upon a standard ARIMA model in terms of forecasting peroformance [51]. Matheson in 2006 has compared the real-time forecasting ability of the Factor Model for New Zealand economy with a variety of other time-series models including the Reserve Bank of New Zealand's published forecasts and found out that the Factor Model performs well particularly at longer horizons[52]. Angelini, Banbura, and Rünstler in 2008 have extended dynamic factor model with cross-equation accounting identities and found out it outperformed quarterly time series models and bridge equations in forecasting GDP growth[53].

However, the factor model has its limitation as well. Indeed factors introduce noise during the process when they summarize and extract useful information from predictors[14]. Besides, factors selected based on the eigenvalues order could not guarantee that they are the factors with high forecasting ability. Evidence lies in the fact that model averaging has beaten models based on factors regrading to the high predictability[54]. Moreover, with the factors as predictors, the forecasting model has been losing its interpreting ability between macroeconomic variables and the GDP growth.

2.4 Bridge Equation

Nowcasting models always need to deal with specific data irregularities: where target variable like GDP growth is sampled at a lower frequency while indicators are available at higher frequencies and in a more timely release. One of the earliest econometric approaches to tackle the mixed-frequency data relies on the application of bridge equation. Introduced by Parigi and Schlitzer in 1995, Bridge Equations are Linear Regression where low frequency variables are on the left of the equation and the explanatory variables on the right side of the equations are quarterly lags of the predictor aggregated over time from the high-frequency observations. A single Bridge Equation with one indicator can be expressed as

$$y_t = \beta_0 + \lambda y_{t-1} + \beta(L)x_t^Q + \varepsilon_t \tag{2.5}$$

where y_t is the GDP growth in quarter t, y_{t-1} is the autogressive term and x_t^Q is the indicator with the same periods as the GDP variable that is available for $t = 1, 2, ..., T_y.\beta(L)$ is the quarterly polynomial indicator of order p which is defined as

$$\beta(L) = \sum_{i=0}^{p} \beta(i+1)L^i \tag{2.6}$$

The predictor x_t^Q is mapped from the high frequency observations to the aggregated low-frequency observations formalized through the deterministic aggregator function $\omega(L^{1/m})$

$$x_t^Q = \omega(L^{1/m})x_t^M = \sum_{j=0}^{r} \omega_j L^{j/m} x_t^M \tag{2.7}$$

The parameters of Bridge equation can be estimated by ordinary least squares (OLS) and with these estimated parameters,a forecast $y_{T_y+h|T_x}$ could be obtained given information up to period T_x

$$\hat{y_{T_y+h|T_x}} = \hat{\beta}_0 + \hat{\lambda}\hat{y_{T_y+h-1|T_x}} + \hat{\beta(L)}\hat{x_{T_y+h|Tx}} \tag{2.8}$$

Bridge Equation (BM) is one of the most popular techniques in short term data nowcasting area due to its convenience in both model estimation and interpretation. Baffigi, Golinelli, and Parigi have estimated BM for aggregate GDP and components both area-wide and for the three main countries of the euro area. It is proved that Bridge Equations is superior than univariate ARIMA, multivariate VAR and structural models in terms of forecasting ability[49]. Bruno, Di Fonzo, Golinelli, *et al.* have examined the forecasting ability of BM for GDP growth in the G7 countries and found BM beat simple ARIMA model and VAR model in forecasting accuracy[56]. Bridge Equations are compared in 2013 with a MIDAS model and mixed-frequency VAR model in their performances of Nowcasting the quarterly growth rate of the Euro area GDP and its components. The results showed that BM model obtained good Nowcasts in general[57]. A GDP Nowcasting project GDPNow, conducted by the Federal Reserve Bank of Atlanta, provides the estimate of real GDP growth in United States by synthesizing the Bridge Equation approach relating GDP sub-components to monthly source day with the Factor Model approach[58].

The disadvantage of BM model lies in that it is a purely statistical model which only includes timely updated regressors. In other word, once the model which exploits the high frequency information is misspecified, the error transmits to the bridge equation and to the nowcasting that are gained recursively[59].

2.5 Mixed Data Sampling Regression

The other typical technique to tackle these data irregularities is Mixed Data Sampling (MIDAS) regression introduced by Ghysels, Santa-Clara, and Valkanov in 2004 which involves

time series data sampling at different frequencies. In MIDAS regressions, the samples of the low-frequency variable are projected directly to lagged high-frequency observations of the predictors without time aggregation[61]. Take GDP nowcasting as an example, quarterly GDP growth denoted as y_t, where t is the quarterly time index $t = 1, 2, ..., T_y$ with T_y being the final quarter for available GDP data. The MIDAS equation for nowcasting GDP growth y_{t+h} in period $t+h$ with a forecast horizon of h quarters is

$$Y_{t+h} = \beta_0 + \lambda y_t + \beta_1 B(L^{1/m}; \theta) x_{t+w}^M + \varepsilon_{t+h} \tag{2.9}$$

where the lead of the high-frequency indicator $w = T_x - T_y$ could indicate more observations of the predictor x than GDP data if $w > 0$. x_{t+w}^M represents the indicators which is available earlier than the current quarter GDP data.

To make the equation simple, Equation 2.9 only contains a constant variable β_0 and one autoregressive term λy_t. In practice the empirical application of more autoregressive lags could be employed[62][18]. The effect of the monthly indicator x_{t+w}^M on y_{t+h} is constructed by the high-frequency lag polynomial $\beta_1 B(L^{1/m}; \theta)$. The polynomial operator is defined as:

$$B(L^{1/m}; \theta) = \sum_{k=0}^{K} b(k; \theta)(j) L^{k/m} \tag{2.10}$$

$$L^{k/m} X_t^M = X_{t-k/m}^M \tag{2.11}$$

where $B(L^{1/m})$ is a polynomial of length j^{max} in the $L^{1/m}$ operator, which produces the value of $X_t^{(m)}$ lagged by j/m periods. If Y_t is the quarterly sample, the equation 2.9 above implies the projection of quarterly Y_{t+h} onto monthly indicator $X_{t+w}^{(m)}$ with up to j^{max} quarterly lags.

To avoid parameter proliferation for long high-frequency lags K, functional lag polynomials are chosen for $B(L^{1/m}; \theta)$[63]. A most common functional form of the polynomial is the exponential Almon lag, with q shape parameters.

$$b(k; \theta) = \frac{exp(\theta_1 k + \theta_2 k^2 + ... + \theta_q k^q)}{\sum_{j=0}^{K} exp(\theta_1 j + \theta_2 j^2 + ... + \theta_q j^q)} \tag{2.12}$$

The MIDAS parameters are estimated for each forecast horizon by non-linear least squares(NLS) and the direct forecast is given by the conditional expectation

$$\hat{y_{T_y + h|T_x}} = \hat{\beta}_0 + \hat{\lambda} y_{T_y} + \hat{\beta}_1 B(L^{1/m}; \hat{\theta}) x_{T_x}^M \tag{2.13}$$

MIDAS has been widely used for financial applications as a macroeconomic forecasting

approach for quarterly GDP. Clements and Galvão have found out that the application of MIDAS regression within-quarter information on monthly indicators can result in marked reductions in RMSE compared with quarterly-frequency AR or AR distributed-lag(ADL) models [64]. Andreou, Ghysels, and Kourtellos have employed MIDAS regressions to handle the mixture of sampling frequencies i.e. matching daily financial data with quarterly macroeconomic indicators in 2013 [18]. Duarte *et al.* have proved that MIDAS regression outperformed benchmarks like simple autoregressive models and tradition quarterly models in terms of forecasting accuracy [62].

Compared with BM model, MIDAS is more robust in the presence of mis-specification. Moreover, it is parsimonious attribute to the lag polynomials which based on a very small number of parameters. Besides, it can be easily estimated by NLS. However, MIDAS approach has a limitation in terms of the missing values in the low frequency variable since it is only possible to obtain a high frequency update of the correlated low frequency realization[59].

2.6 Limitations of the Econometric Approach

Although the Econometric approach for GDP Nowcasting has dominated the forecasting area for many years, there is never a perfect model to get the estimated GDP growth. Every model has its limitations and weaknesses.

The Univariate Autoregressive forecasts like ARIMA and Random Walk model approximate estimated value only relying on the historical value of GDP growth, which decline the possibility of modelling the complex system that an economy is. Vector Autogressive models, as a natural extension of the Univariate Autoregressive Model can create structural equations including a limited number of leading indicators for the economy but the VAR model is not capable to predict GDP from a large vector of indicators. Furthermore, Dynamic Factor Models summarize the information available into a small number of factors while factors selected based on the eigenvalues order could not guarantee that they have high forecasting ability. The disadvantage of Bridge Equations is that once the high frequency information is misspecified, the error transmits to the equation and to the Nowcasting that are gained, recursively. In comparison, the limitation of MIDAS lies in the missing values in the low frequency variables since the high frequency updates is obtained from the low frequency variables.

Most importantly, the traditional time-series models used in forecasting typically only incorporate a small handful of variables, obtained from a variety of different selection procedures. The final variables selected are thus considered representative of a larger population of potentially useful series[52]. This would cause problems of statistical con-

sistency and degree of freedom in complex models requiring lots of explanatory variables. The surge of large number and different types of data with the arrival of "big data" makes more related information available to increase forecasting accuracy. At the same time, it has raised the requirement for forecasting models to handle a large amount of unbalanced variables. Traditional econometric models have often been asymptotically estimated, the marginal information gain thus decreases with the sample size increasing to a certain size[65]. The decreasing prediction efficiency for Econometric Approach dealing with large datasets creates the need for a more powerful and "data-rich" method.

Chapter 3

Machine Learning Approach for GDP Nowcasting

In the big data era, the rapidly increase of high frequency data both in the number of features and the number of instances has raised a requirement for a prediction model to consider the trade-off between accuracy and efficiency. With the improvement of computational ability, the data-driven approaches like Machine Learning Methods have become widely used and shine in many applications. The two key elements that drive this success are the appropriateness of statistical models which extract the sophisticated data dependencies and the scalability of the learning system that capture the model of interest from the training dataset[66].

Recent advances in Machine Learning have shed new light on applied econometric[67]. The field of Machine Learning could be summarized as "the model learns by precious experience", if it improves on its performance doing a task[68]. This broad definition emphasis the interdisciplinary nature of Machine Learning and presents its relevancy in economics. When dealing with optimization problems, Machines learn from underlying patterns and improve with experience.

With proper related data, the learning algorithm recurrently adjusts its parameters to minimize the error between the target variable and the predicted output which points out the main difference from an econometric approach. Traditionally, Econometric and Machine Learning aim to solve different types of problems, and have separate development. The Machine Learning Algorithms, which focused more on prediction, are built aiming to improve the prediction of output, while the traditional econometric approach focused on explanation, often revolving around parameter estimation [69]. However, the difference between these two fields reach an agreement in the field of Nowcasting, where the issues of causality and economic interpretability are less relevant. Recall, the goal of GDP Now-

casting is to extract a signal from a broad range of noisy higher-frequency indicators which could reflect current economic environment. It does not matter whether the indicator is a causal factor that shapes GDP or it is just a symptom of economic growth[70].

Besides its strong forecasting ability in handling large number of predictors, the other advantage of Machine Learning techniques as a "data-rich" approach over Econometric Models, is their ability to handle non-linear data structure and its interactions. The form like tree based algorithms is more flexible compared with Econometric Techniques and could be more adaptive to data [71]. In this Chapter, the theory of the Machine Learning Algorithms applied in our research is introduced.

3.1 Linear Methods

Strictly speaking, Linear Methods like Linear Regression should be considered as traditional statistical models rather other Machine Learning Methods because of their linear structure. However, the distinction between statistics and Machine Learning is often blurry and the linear algorithm is the basis of any rudimental Machine Learning Algorithm. In this research, more attention is paid to the shrinkage method of Linear Models like Ridge Regression and Lasso.

3.1.1 Linear Regression

Consider the standard model of multiple linear regression, the model predicts output with a linear combination of input features. As shown in the equation 3.1, x_i is the ith input data vector including constant coefficient w_0 while w stands for weights or coefficients matrix.

$$f(x_i) = \sum_{m=1}^{P} w_m x_{im} + w_0 = w^T x_i \tag{3.1}$$

Linear regression looks for optimizing w to minimize the residual between estimated outcome and the real value. The most common way to approach this target is the Least square method, whose loss function can be written as Equation 3.2 where $\|.\|$ denotes the L2 norm, y is the target variable matrix, X is the input variable matrix and w represents the coefficient matrix.

$$L(x) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}\|y - Xw\|^2 \tag{3.2}$$

The extreme point of the loss function could be found by taking the derivative of w in Equation 3.2 and make it equal to zero. In this process, the optimal solution of coefficient

matrix w is obtained as Equation 3.3, where X is the input matrix and X^T is the transpose matrix of it, y stands for the target variable matrix.

$$w^* = (X^T X)^{-1} X^T y \tag{3.3}$$

The estimation procedure is unbiased and have minimum variance under the condition that $X^T X$ is invertible. When the feature number p is bigger compared with the prediction instances n (fat regression), or there is strong correlation between features (multi-collinearity), which is quite common in the real-world challenges, the least square estimates are sensitive to a number of "errors" and do not make sense[72]. Over-fitting in linear regression, i.e. low bias but large variance, is another problem that makes the ordinary least squares estimation less satisfiactory in real life.

It is proved that shrinking some coefficients could somehow sacrifice a little bias to reduce model complexity and hence improve the overall prediction accuracy[73]. Besides, the regularization process of w guarantee the invertibility of $X^T X$ matrix. The most widely used shrinkage methods for linear regression are ridge regression and Lasso.

3.1.2 Ridge Regression

In Ridge Regression, the magnitude of the coefficients are constrained by adding a penalty. The loss function is expressed in Equation 3.4.

$$L_R(w) = \|y - Xw\|^2 + \lambda \|w\|^2 \tag{3.4}$$

where y denotes the target variable matrix, X denotes the input matrix, w stands for coefficient matrix and λ is the penalty term regularizes the coefficients. Setting λ from 0 to 1 controls the power of penalty, the larger its value, the stronger the coefficients' size is penalized.

The optimal solution of coefficient is expressed as Equation 3.5, where I denotes the identity matrix.

$$\hat{w_R} = (X^T X + \lambda I)^{-1} X^T y \tag{3.5}$$

Ridge Regression is perfectly sensible in circumstances in which the large weight or coefficients are unrealistic from a practical point of view and it works specifically on solving multi-collinearity. However, the major limitation of Ridge Regression is that the choice of the biasing constant λ is a judgemental one, users should be careful in selecting the value of λ since it decides the accuracy of the prediction model directly.

As the state-of-the-art strategy to reduce the computation complexity, Ridge Regres-

sion has gained excellent learning performances in many practical applications[74], provided the number of predictors m is not very big. Some researches have asserted that the range of m depends on the regularity of the regression function [75] [76] [77]. This key problem results in users selecting only a small m or taking m as a parameter in the learning process.

Kernel Ridge Regression is the Kernelized Ridge Regression by introducing unlabeled data, i.e. inner products between training examples in the learning process. The data needs to be re-divided again and again and it requires a large amount of communication by taking m as a parameter. Kernel Ridge Regression avoids this process for a large range of m. With enough inner products given, the optimal learning rate can be achieved. It implies feature space with computing the inner products between the data instead of computing the coordinates of the data which greatly improve the efficiency.

In other words, with a certain "kernel trick", Kernel Ridge Regression could transform linear regression into a non-linear and high-dimensional feature space[78]. The data x_i in the input dataset X is replaced with the feature vectors: $x_i \rightarrow \Phi = \Phi(x_i)$ operated by the kernel where $K_{ij} = k(x_i, x_j) = \Phi(x_i)\Phi(x_j)$. Kernel Ridge Regression Methods are employed in this study with different kernel functions including linear kernel, rbf kernel, Laplacian kernel and so on.

3.1.3 Lasso

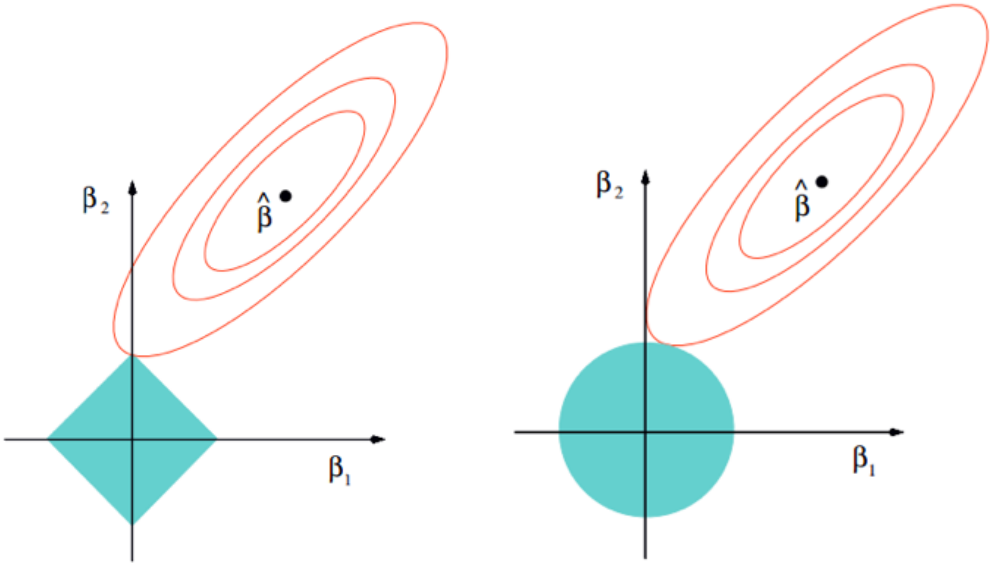
Besides the prediction accuracy, the interpretation of a model should also be taken into account. With a large scale of input dataset, we should focus more on the small subset of predictors with the strongest effects. Although Ridge Regression shrinks coefficients and is proved to be stable, it does not make any coefficient to 0. Lasso stands for Least Absolute Shrinkage and Selection Operator is an algorithm similar to Ridge Regression. It also adds a penalty for non-zero coefficients, but unlike Ridge Regression which applies an L2 penalty, Lasso uses the sum of their absolute values, i.e L1 penalty. As a result, it retains both the prediction accuracy and feature selection function [73]. The loss function of Lasso is expressed as equation 3.6.

$$L_L(w) = \|y - Xw\|^2 + \lambda \|w\| \tag{3.6}$$

In this case, the gradient is not defined, as the absolute function is not differentiable at $x = 0$. Lasso uses a different technique called coordinate descent to solve this problem. The optimal solution of w is expressed as Equation 3.7.

$$\hat{w_j} = \begin{cases} g(w_{-j}) + \frac{\lambda}{2}, & if g(w_{-j}) < -\frac{\lambda}{2} \\ 0, & if -\frac{\lambda}{2} \leq g(w_{-j}) \leq \frac{\lambda}{2} \\ g(w_{-j}) - \frac{\lambda}{2}, & if g(w_{-j}) > -\frac{\lambda}{2} \end{cases} \tag{3.7}$$

Here $g(w_{-j})$ represents the approximate difference between actual outcome and the predicted outcome considering all the input variables except the jth. If this value is small, it means that the algorithm is able to predict the outcome fairly well even without the jth variable and thus it can be removed from the equation by setting a zero coefficient.



(a) lasso (b) ridge regression

Figure 3.1: Estimation Picture For The Lasso And Ridge Regression

Source: Course Notes for Predictive Modeling, MSc in Big Data Analytics at Carlos III University of Madrid

Figure 3.1 gives a rough idea of the difference between Lasso and Ridge Regression. Assume a hypothetical data-set with only two features. Using the constraint for the coefficients of Lasso and Ridge, the optimal coefficients are determined by finding the first point where the elliptical contours hit the region of constraints. It is obvious that the solution of Lasso occurs on the axes, i.e, one of the features completely vanishes. For a higher dimensional feature space, Lasso keeps only the important features while regularizing the linear model.

3.2 Support Vector Machine

Support Vector Machine (SVM) was introduced by Boser, Guyon, and Vapnik in 1992. Support vector machine is a specific supervised learning algorithm for classification and regression with the prominent characteristics of the capacity control of the decision function, the use of the kernel functions as well as the sparsity of the solution [79][80]. SVM has shown remarkable model ability to resist the over-fitting problem i.e. achieving a high model generalization performance by its unique principle of the structural risk minimization principle to estimate a function by minimizing an upper bond of the generalization error. Besides, the solution of SVM is a linearly constrained quadratic programming problem that ensures the estimation results are always unique and globally optimal[81].

The idea of SVM for regression is to generate a hyperplane in the D-dimensional feature space that approximate $h(x)$, which is the target time series. The data set is given as $G = (x_i, y_i)_{i=1}^{N}$. The SVM considers approximating functions of the form based on the knowledge of G:

$$f(x, w) = \sum_{i=1}^{D} w_i \varphi_i(x) + b \tag{3.8}$$

where $\varphi_i(x)$ are the features and w_i are the coefficients which are estimated by minimizing the following function:

$$R(C) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i, w)|_{\varepsilon} + \lambda ||w||^2 \tag{3.9}$$

where λ is a constant term and $|y_i - f(x_i, w)|_{\varepsilon}$ is the robust error function which defined as:

$$|y_i - f(x_i, w)|_{\varepsilon} = \begin{cases} 0, if\, |y_i - f(x_i, w)| < \varepsilon \\ |y_i - f(x_i, w)|\,, otherwise \end{cases} \tag{3.10}$$

The method to minimize Equation 3.9 is determined by a finite number of parameters with the following form:

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(x, x_i) + b \tag{3.11}$$

where $\alpha_i^* \alpha_i = 0, \alpha_i, \alpha_i^* \geq 0 \;\; i = 1, ..., N$, and $K(x, y)$ is the kernel function, which describes the inner product in the D-dimensional feature space, designed to solve non-linear problems.

Thanks to the advantages of SVM algorithm in solving non-linear problems and its

exceptional resistant to over-fitting, it has been widely used in the time series forecasting area. Mukherjee, Osuna, and Girosi in 1997 have implemented SVM on the data base of chaotic time series and obtained better performance than the Mackey-Glass time series[82]. Tay and Cao have examined the predictability of SVMs and BP networks for five time series in 2001 and showed that SVM outperformed BP networks on the criteria of MSE and MAE [83]. Kim have applied SVM to the stock price index forecasting and found out SVM to be a promising alternative to stock market prediction[84].

Notably, the success of SVM to provide a close approximation highly relied on the proper selection of the hyper-parameters such as the kernel parameter and the constant term. An improper choice of these parameters may lead to totally biased results.

3.3 K-Nearest Neighbor Regression

The K-Nearest Neighbor (KNN) algorithm has been used extensively for both classification and regression based on a feature similarity techniques. The KNN classifier evaluates a testing point based on a fixed number (k) of neighbor points whose classification are known in the feature space[85]. When comes to regression, the KNN algorithm assumes that similar patterns in the past would emerge in the future sequences[86] and generates estimation of the response value of testing point as a weighted average of response value of the k nearest neighbour training points. A kernel trick is applied to compute the weight of each referral point based on its proximity to the testing point.

$X = \{x_1, x_2, ...x_M\}$ represents M training points which possess N features. The distance between each training point x_i and the testing point x_t can be calculated with weighted Euclidean distance, shown as

$$d(x_t, x_i) = \sqrt{\sum_{n=1}^{N} w_n (x_{t,n} - x_{i,n})^2} \tag{3.12}$$

where N is the number of features, $x_{t,n}$ and $x_{i,n}$ denote the testing point x_t and the ith training point x_i, w_n is the weight between 0 to 1 assigned to the nth feature based on its feature importance.

Then, the k closest training points could be selected as the k nearest neighbors of x_t. The illustrative presentation of the k nearest neighbors in the two-dimensional feature space is shown in Figure 3.2, where the three triangle points with the smallest Euclidean distances are selected as the k (=3) nearest neighbor training points and the other training points whose distance with the testing point is larger than those are treated as non-neighbour points.

The Kernel trick has been applied to approximate the non-linear characteristic of the

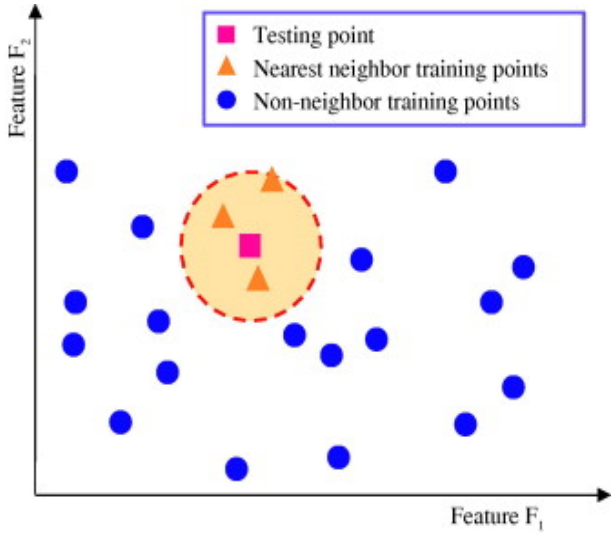


Figure 3.2: K-Nearest Neighbors of A Data Point in A Two-Dimensional Feature Space

Source: Hu, C., Jain, G., Zhang, P., Schmidt, C., Gomadam, P., and Gorka, T. (2014). Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery. Applied Energy, 129, 49-55.

testing data series. The response of x_t is expressed below with kernel regression:

$$\hat{f(x_t)} = \frac{\sum_{i=1}^{k} \phi(x_t, x_(i)) f(x_{(i)})}{\sum_{i=1}^{k} \phi(x_t, x_(i))} \tag{3.13}$$

where k refers to the number of nearest neighbors used for estimation, $\phi(x_t, x_{(i)})$ is the kernel function employed to the ith training data point $x_{(i)}$ and $f(x_{(i)})$ is the known target value of $x_{(i)}$. The parameter k acts a smoothing role to adjust the flexibility of KNN regression model. In this paper, radial basis function (RBF) kernel has been used in KNN regression, expressed as

$$\phi(x_t, x_{(i)}) = e^{-\frac{d(x_t, x_{(i)})^2}{2\sigma^2}} \tag{3.14}$$

where $d(x_t, x_{(i)})$ is the Euclidean distance between the training data point $x_{(i)}$ and testing point x_t, and σ is a Gaussian decay factor which controls the exponential function of the difference between two data points[85].

Due to its intuitively appealing nature and competitive theoretical properties, KNN regression has won considerate attention in time-series applications[87].

In the community of statistics, Yakowitz in 1987 has introduced a non-parametric regression device like the kernel method to horologic data and extended the pattern recognition broader[87]. Cleveland has enhanced the visual information on a scatter-plot by

applied a robust locally weighted regression method[88].

In the financial time series forecasting field, a research in 2003 has compared the prediction ability of KNN regression and a risk-adjusted buy-and-hold strategy as well as a linear ARIMA-based model to the New York Stock Exchange. The results showed that KNN regression is superior to the other methods in terms of returns for all the years studied and yielded higher sharp ratios[89]. Meade examined this empirically, with different forecasting methods and different frequencies of foreign exchange rate behaviour, including a linear AR-GARCH model and four non-linear methods, namely, three nearest neighbour methods and locally weighted regression. The results showed no evidence of any significant difference between the non-linear generating process and the linear model[90].

3.4 Tree Based Modeling

Statistical models like linear regression have dominated forecasting area for many years. However, these models have their intrinsic pre-defined model assumptions which sets a limitation to the relationship between dependent variables and target variables. Tree-based models like Decision Tree, Random Forest and Adaboost, act as non-parametric model which do not require the underlying relationship between regressors and the predictors and have been shown powerful model ability dealing with both prediction and classification problems[91].

3.4.1 Regression Tree

The Decision Tree is the fundamental structure of tree based models, which having its origin in machine learning theory, provides an effective solution for classification and regression problems. When the predictor variable is discrete valued, a classification tree is generated, whereas a regression tree is built for a continuous target variable.

A regression tree determines the predictive value based on a series of questions and conditions. The data space constructed by all the training samples is split into distinct and non-overlapping partitions in an iterative process. The splitting strategy is to minimize the sum of the squared deviations from the mean in the separate parts. The process continues until the number of training samples at the branch reaches a user-specified minimum node size. To the testing point falls into a region, the algorithm makes the prediction with the mean of response in the training set in that particular region. The Figure 3.3 shows the tree growing process.

Since the tree construction is built from a training sample, the full structure might suffer from over-fitting that performs poorly on model generation to new testing data points. Thus, pruning is applied to set the user-specified cost complexity factor which minimize

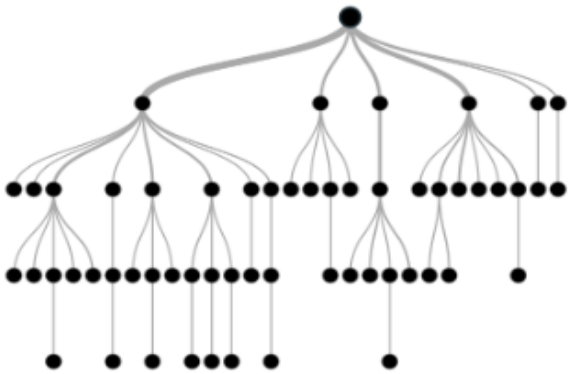


Figure 3.3: Decision Tree Growing Process

Source: Panda, S., Pattanaik, P. A., and Swarnkar, T. (2017). A Higher Education Predictive Model Using Data Mining Techniques. In DIAS/EDUDM@ ISEC.

the sum of the output variance in the validation data. The pruning process governs the tradeoffs between tree-size (model bias) and its goodness of fit (model variance)[92].

The principal advantage of Decision Tree over other modeling techniques is that it equips easily understood interpretable rules or logic statements. The explanation capability lies in trees producing axis parallel decision surface. Besides, it performs without complicated computation and provides clear information on factor importance for prediction and classification [93]. However, when compared with neural networks for nonlinear data, decision tree induction generally performs worse and it shows a susceptibility to noisy data[94]. To sum up, a decision tree is more effective in solving a classification problem, and unless evident data trends or sequential patterns are shown, this technique is less appropriate for forecasting time series data[93].

3.4.2 Random Forest

It is easily understood that a single Decision Tree suffers from bias and variances and that ensemble learning could significantly improve this problem. Ensemble learning is a model generally make predictions by referring to a set of results from different models. By combining individual models, the Ensemble Model performs more flexibly with less bias and less data sensitivity.

The Tree-Based Ensemble Learning Algorithms like Random Forest, Adaboost and Gradient Boosting are the combined prediction results with multiple decision trees in different organization forms. Two typical Ensemble Methods are Bagging and Boosting.

- **Bagging:** Training a bunch of individual models in a parallel way. Each model is

trained by a random subset of the data

- **Boosting:** Training a bunch of individual models in a sequential way. Each individual model learns from mistakes made by the previous model.

A Random Forest is a collection of Decision Tree prediction using Bagging as the ensemble method[95]. The un-weighted average process is expressed below:

$$\bar{h(x)} = (1/K) \sum_{k=1}^{K} h(x; \theta_k) \tag{3.15}$$

where $h(x; \theta_k)$ denotes the prediction result of individual decision tree k, K is the total number of decision tress and $\bar{h(x)}$ represents the estimated response value of the testing data point.

As an adaptive Tree-Based Ensemble Learning, Random Forest has many desirable features. First, the Random Forest method requires few parameters to tune. Second, Random Forest could generate an out-of bag error in its growing procedure while non-tree based models require extra training procedures like cross validation to reach similar estimates. Third, Random Forests provide variable importance and easy model interpretation. Forth, applying a Bagging strategy, compared with the single method, Random Forest is robust in model generalization ability[96]. Fifth, Random Forest tends to be easier to adapt with more data by growing more branches to expand itself[97].

The main limitation of Random Forest is the ineffectiveness for real-time prediction caused by the computational complexity to generate large number of tress. Besides, similarly with other tree-based models, Random Forest is a predictive modelling tool rather than a descriptive tool, so that it is difficult to describe the relationship between independent variables and the explanatory variable.

There are a lot of application of Random Forest Regression in the forecasting area: Mei, He, Harley, *et al.* investigated the real-time price forecasting in New York electricity market with Random Forest and found that Random Forest method to have accurate forecasting ability and strong model adaptability and outperformed existing price forecasting methods [96]. Creamer and Freund compared the performance of Random Forest and Logistic Regression on predicting corporate governance risk in Latin American markets. Random forest was found to be superior than Logistic Regression[98]. Van den Poel and Lariviere estimated both customer retention and profitability outcomes with Random Forest, Ordinary Linear Regression and Logistic Regression models. They demonstrated that Random Forests provide better model fitness[99].

3.4.3 Adaboost

Adaboost is an adaptive Boosting Ensemble Model that learns from the misclassfication data points by increasing the weight of them.

The main idea of the Adaboost algorithm is an iteration process: firstly, initialize the weight of each data point with equal importance; next, a Decision Tree (weak classifier or regressor) is trained and its weighted error rate is calculated. The weighted error rate is the percentage of wrong predictions with their corresponding data weight out of total total predictions. Then, the weight of each Decision Tree is calculated based on the its weighted error rate. The one with higher error rate is entitled with less decision power during the last voting. After that, the weight of each data point is updated following the rule that the weights of misclassified data points are exaggerated while the weights of the correct predictions stay the same. Last, the iteration process is repeated until the number of trees set beforehand is reached and the final prediction is made in the last round. By updating the weight of each data point and the weight of each decision tree, the last prediction is guaranteed to have the least error rate. Each decision tree acts as a weak classifier whose prediction ability is only required to be stronger than a random guess, and the combined result is proved to have an exponential loss[100].

The application of Boosting to a regression problem is to reduce regression problems to classification ones and follow the Boosting procedure [101]. Avnimelech and Intrator have modified the original Adaboost algorithm to forecast the laser data in the Santa Fe time series competition in 1999 [102]. Goh, Lim, and Peh have predict drug dissolution profiles for developing drug dosage regimens with the modified AdaBoost method [103]. Canestrelli, Canestrelli, Corazza, *et al.* have applied AdaBoost into forecasting of tide levels in 2007[104]. Heo and Yang have analyzed the prediction ability of Adaboost and other algorithms like ANN, SVM and decision tree to forecast the financial risk for Korean construction companies and proved Adaboost has more prediction power than others[105].

Although the studies aforementioned have shown Adaboost can be effective, it suffers some drawbacks:

First, this method is sensitive to outliers and noisy data. Due to the updating principle of AdaBoost, these examples with large prediction errors are strongly boosted while the one with smaller errors are weakly boosted which might result in the low overall performance of the committee machine[106].

Second, Adaboost regression works on expanding each regression observations into classification instances. Even if the integral above is generated piece wisely linear, the number of these pieces could grow linearly in the number of boosting iterations[107].

Third, except the first iteration, the loss function that each weak regressor minimizes is not standard. Moreover, the different loss function between iterations and even between

examples on the same iteration makes it difficult to examine whether a specific base regressor is appropriate for the whole model[107].

To improve these shortcomings of Adaboost, a gradient descent approach i.e Gradient Boosting has been introduced.

3.4.4 Gradient Boosting

Among all the application of machine learning methods in the real word, Gradient Tree Boosting has proven to have state-of-the art performance in many learning tasks including classification, regression and ranking[108].

Compared with Adaboost, Gradient Boosting is another boosting model learns from the residual error directly rather than update the weights of data points. The residual errors are the gradient of the loss functional being minimized with respect to the model values at each training data point evaluated at each iteration. Besides, a subsample of the training data is selected randomly from the full training data set at each iteration, this randomization process is proved to improve the approximation accuracy and execution speed of gradient boosting as well as the robustness against overcapacity of the base learner[109].

The process of how Gradient Boost learns is demonstrated. Firstly, a Decision Tree is trained and applied to predict the testing data points. Then the residual of this Decision Tree is calculated and saved as the new target variable. Repeat the Decision Tree growing step until the user-specified tree size is reached. The final prediction is made by simply adding up the predictions of all the trees generated.

Due to its powerful forecasting ability to capture complex non-linear function dependencies, Gradient Boosting has been widely used in recently years. Gradient boosting has been applied for the deterministic forecasting of solar power in the Global Energy Forecasting Competition 2014[110]. Zhang and Haghani in 2015 have employed a Gradient Boosting Regression Tree Method to analyze and model freeway travel time and have proved that the prediction accuracy and model iterpretability were improved considerably[111]. A study of pattern recognition of waste generation in 2017 has built a Gradient Boosting Regression to predict weekly waste generation in New York City[112].

Although the Gradient Boosting machine has shown considerable success in various practical application, it still has some drawbacks. The most prominent problem of Gradient Boosting is its memory-consumption. The cost of storing for a predictive model is determined by the number of boosting iterations. In some accuracy intensive tasks, the desired number of iterations can be the range of tens of thousands, which could increase the time requirment not only in the training but in the evaluation session. Besides, the learning procedure of Gradient Boosting is essentially sequential and problematic with

parallelization by design[113].

These problems mentioned are generally computational and are improved by Xgboost and Lightgbm in the last few years. Both Xgboost and Lightgbm are implementations of a generalized gradient boosting algorithm. Xgboost has won a reputation as one of the most powerful and scalable algorithms in many machine learning competitions on platforms like Kaggle, while with its faster training speed and equal or even higher efficiency than XGboost, Lightgbm has gained more and more popularity.

3.4.5 Xgboost

Xgboost stands for extreme Gradient Boosting, compared to other Gradient Boosting models, Xgboost uses a more regularized model formalization to control over-fitting. Instead of optimizing plain squared error loss, the Xgboost algorithm introduced a regularized objective function $\omega(f)$ to smooth the final learned weight [114][115][66]. Therefore, the objective function of Xgboost can be expressed as:

$$Obj = \sum_{i} l(y_i, \hat{y}_i) + \sum_{k} \Omega(f_k) \tag{3.16}$$

Here $l(y_i, \hat{y}_i)$ denotes any differentiable convex loss function that measures the difference between the estimation and ground truth for a given training instance. $\omega(f_k)$ is the penalty term controlling the complexity of the model. It is a function of "leaf number T" and "leaf node output result W_j in each sub-decision tree. It is defined as follows [66]:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} W_j^2 \tag{3.17}$$

where T stands for the number of leaves of tree f_t and w is the leaf weights, the score in the corresponding leaves constitute the final predictions.

The Tree Based Ensemble Model in Formula 3.17 includes functions as parameters that cannot be trained with traditional optimization methods in Euclidean space. This problem is solved by an iteration path. As the iterative learning process of the algorithm continues, the loss function gradually decreases, while the regular terms continue to expand. Expanding the objective function into Taylor second order series, the weak trees minimize the objective function which could be found by taking the second derivative. It is expressed in the form shown below:

$$f_t(x)^* = -\frac{\sum_{i \in I_t} \partial_{ij(t-1)} l(yi, \hat{y_i}^{(t-1)})}{\sum_{i \in I_t} \partial^2_{ij(t-1)} l(y_i, \hat{y_i}^{(t-1)}) + \lambda} \tag{3.18}$$

Here, I_t denotes the counting range of the iteration, set as an algorithm parameter in the beginning. The algorithm that starts from a single leaf and splits into branches to the tree iteratively. The model automatically recognizes and stops the iteration when the number of variables involved exceeds maximum value. The target decision tree grows on the layer to decrease the loss function while the regularized function removes the redundant branches during the iteration. In the end, the final decision tree model with the minimum objective function could be gained under equilibrium condition for both classification and regression tasks [115].

Despite its excellent regularization performance in controlling over-fitting, the most important attribute that led to the success of this algorithm is its excellent scalability in all scenarios. It has been proved to be more than ten times faster than existing popular solutions on a single machine and extends to billions of cases under distributed settings or settings with limited memory[66]. This owes to some unique features as follows:

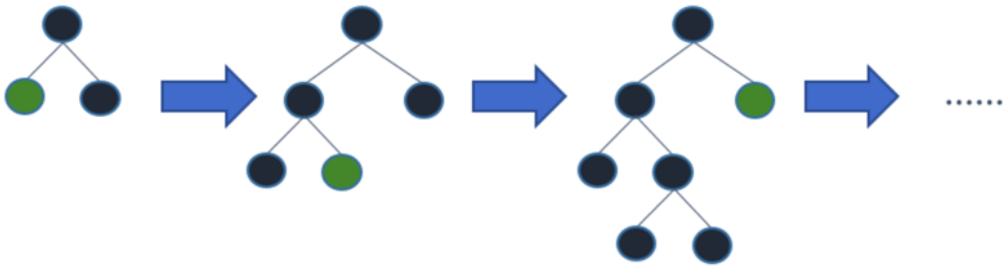
- **Sparsity-aware Split Finding.** Missing values, zero entries and artifacts of feature engineering such as one-hot encoding in the real-world problems often makes input dataset to be sparse. It is crucial to inform the algorithm of the sparsity pattern in the data. Xgboost added a default direction in each tree node which helps the model recognize missing values and learn to handle it.

- **Weighted Quantile Sketch.** Most existing tree based algorithms can propose candidate split points when the data points are of equal weights, i.e. using quantile sketch algorithm. However, it does not apply to weighted datasets. Xgboost introduced a novel distributed weighted quantile sketch algorithm that can solve this problem with a provable theoretical guarantee.

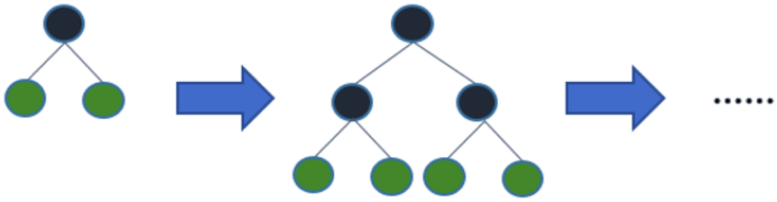
- **Column block for parallel learning.** Xgboost reduces the time requirement of sorting data into the right order by storing the data in in-memory units called blocks. Unlike other algorithms, this enables the data layout to be reused by subsequent iterations, and it only needs to be computed once before training. This feature also supports a linear scan of the split finding process and column sub-sampling, which significantly improved the computation speed.

- **Cache awareness.** In Xgboost, non-continuous memory access is required to get the gradient statistics by row index. Hence, Xgboost has been designed to make optimal use of hardware. This is done by allocating internal buffers in each thread, where the gradient statistics can be stored.

- **Out-of-core computing.** This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory.



(a) Leaf wise tree growth



(b) Level wise tree growth

Figure 3.4: Comparison between Leaf Wise and Level Wise Tree Growth

Source: Liu, L., Ji, M., and Buchroithner, M. (2017). Combining partial least squares and the gradient-boosting method for soil property retrieval using visible near-infrared shortwave infrared spectra. Remote Sensing, 9(12), 1299.

As described by one of the creators Tianqi Chen, the name of Xgboost represents the engineers' goal to extend the computation limit for boosted tree algorithms. It does realize the developers' wish, Xgboost has been widely recognized in a number of machine learning and data mining competitions. It has kept the record to win both the first and second place for machine learning challenges host by Kaggle in 2016 and 2017. According to Lei, Chen, Liu, *et al.*'s experiment in 2017, Xgboost has outperformed the other five machine learning approaches in the accuracy of prediction chemical-induced respiratory toxicity. Similarly, Xgboost has beaten Gradient Boosting model, Random Forest Regression, Extreme Gradient Boosting model in the global and local wind energy forecasting[117]. Torres-Barrán, Alonso, and Dorronsoro also proved in a power load prediction that the Xgboost prediction model has excelled over Random Forest, Bayesian and the K-Nearest Neighbors models in both speed and accuracy.

3.4.6 Lightgbm

Although Xgboost has dominated the machine learning challenges in Kaggle for while, a new contender released from Microsoft in 2017 called Lightgbm has been claimed to be faster with comparable accuracy when considered with the same datasets. Lightgbm is prefixed as "light" for the reason of high speed. In other word, it can handle the large size of data and occupies less memory to run. There are two main novel techniques in Lightgbm that work towards this goal, Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bunding (EFB).

Gradient-based One-Side Sampling. For Gradient Boosting Decision Tree (GBDT) algorithms, the key part and the most time-consuming process is to find the best split points in growing a decision tree[118]. Xgboost employs the pre-sorted algorithm to find split points. This method is a greedy algorithm which enumerates all possible split points on the pre-sorted feature values, which correspondingly, is inefficient in both training speed and memory consumption[119][120]. Another popular method is the Histogram-Based algorithm, which classifies continuous feature values into discrete bins and uses these bins to build feature histograms during training. Since bins are much smaller compared the whole dataset, histograms building outperformed in the computational complexity[121][122][123]. Lightgbm develops GOSS based on this.

In GBDT, there are no native sample weights which could serve as an indicator for the importance of data instances as in Adaboost. Gradient-based one-side sampling uses the gradient for each data instance as a weight to split the data. All the instance with large gradients are kept while instances with small gradients are randomly sampled to keep the distribution of the dataset. By doing so, Lightgbm achieves a good balance between reducing the number of data instances and keeping the accuracy for learned

decision trees[118]. When reflected into the way of growing the tree, Lightgbm employs the leaf-wise strategy while most GBDT methods use a level-wise strategy. As shown in the Figure 3.4a and Figure 3.4b,the level-wise strategy splits every node in the certain level whereas the leaf-wise strategy selects the leaf with max delta loss to grow which can reduce more loss than the level-wise method.

Exclusive Feature Bunding. Lightgbm proposes a novel method in reducing the feature space dimension. The large-scale datasets in real word applications are usually very sparse. The sparsity of the feature space allows the possibility to reduce the feature numbers. Xgboost deals with data sparsity by ignoring the features with zero values. However, histogram-based algorithms does not have efficient sparse optimization solution without retrieving feature bin values. Lightgbm bundles these mutually exclusive features which never take nonzero values simultaneously into a single feature (exclusive feature bundle). Since the number of bundles are way less than the number of features, Lightgbm significantly speeds up the training process without hurting the accuracy[118].

Moreover, the other advantage of Lightgbm lies in the treatment of categorical features. Like Random Forest, Xgboost cannot manage categorical features by itself, it only accepts numerical values. Therefore one has to perform various encodings like label encoding, mean encoding or one-hot encoding before supplying categorical data to Xgboost while histogram-based Lightgbm enables itself to handle categorical features by taking the input of feature names. It does not convert to one-hot coding, and is much faster than one-hot coding. Lightgbm also supports GPU learning which made it widely welcomed among data scientists. The only limitation of Lightgbm is that it's very sensitive to overfiting and is not advisable to apply on small datasets.

3.5 Model Averaging

An Ensemble Method is a wildly used Machine Learning Method to combine the result of a set of weak classifiers by taking a (weighted) vote of their predictions. In this paper, eleven Machine Learning Methods have been employed to our data sets. Given the basis that each algorithm has a variety combination of parameters, and the model generalization of each result is uncertain, a model of averaging method i.e. Ensemble Method is used to mitigate the risks. The motivation to propose the ensemble method comes from the following perspectives:

- **Model Selection Problem in The Real World.** In practice, there is almost no certain answer to determine whether a specific method is more effective than the other in Out-of-Sample forecasting. Therefore, a researcher generally applies a set of different models and chooses the one with the most accurate result when

selecting the right technique for their particular situations. However, the model's prediction ability to a great extent relies on many other influencing factors like sampling variation, model uncertainty and data structure. The model selected with the best performance is not necessarily good for future implementation. The main strength of Ensemble Methods is that they are often much more accurate and steady than the individual decision made by the weak classifiers[124]. In other words, the Ensembles make up the individual algorithm and constitute a stronger classifier or regressor. The uncertainty of the model selection problem could be effectively reduced by combining different methods with little extra effort.

- **The Complex Data Pattern Problem.** Time series data in the real-world is hardly pure linear or nonlinear. They are often a combination of linear and nonlinear patterns. In this way, a single linear or non-linear model is not adequate to forecast the time series well since it alone could not capture the complex data structure well. Hence, by combining both linear and non-linear machine learning algorithms in this study, it guarantees better accuracy of modelling complex auto-correlation structures in the data.

- **The Ensemble Method effectively Improves the Forecasting Accuracy.** It is universally recognized in the forecasting area that no single model is perfect for every problem[125][126][127]. Due to the complex and changeable data structure in the real-world problem, no individual model could be able to capture different patterns at the same time. Empirical evidences have shown that forecasting accuracy could be improved through an ensemble method over a single method without finding the "true" or "best" model[128][129]. This could be explained by the function of an Ensemble Model to incorporate each model's unique feature for different data pattern and balance the limitation of single method for complex data structure[130]. For example, algorithms like KNN make predictions relying on the neighbour data points might be too conservative to estimate the economic growth in a transition period while models like SVM in some cases are too radical with non-linear kernels. Besides, the forecasting robustness is likely to be enhanced by the process of combining several single algorithm.

The combination strategy in this study relied on weighting the estimations of different prediction methods and summarizing them into a single prediction. As shown in Equation 3.19, individual weights are assigned to all models based on their prediction performance

measured by mean square error (MSE).

$$W_i = \frac{\frac{1}{MSE_i}}{\sum_{i=1}^{N} \frac{1}{MSE_i}} \tag{3.19}$$

Mean squared error is a measurement of accuracy for a forecast statistic model. It calculates the difference between predictions and samples and averages the squared value, returning a single number. MSE is defined as Equation 3.20:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 \tag{3.20}$$

where Y_i is the real values of a time series sequences and $\hat{Y}_i$ stands for the predicted values of each algorithm.

The smaller the MSE value is, the closer the predicted values and the real values. Using inverse-MSE weights derived from their sum of squared errors to ensemble all the results will minimize a quadratic loss function based on prediction errors[131].

Chapter 4

Methodology

In this Chapter, an introduction of the methodology of the empirical experiment is presented. It contains the tools used to conduct the experiment with, the workflow of experiment implementation, basic information about the dataset and variables selection, data interpolation and feature engineering process, the theory of data shifting trick, as well as the evaluation matrix for model comparison.

4.1 Tools

4.1.1 The Jupyter Notebook

The Python environment used to process data is Jupyter Notebook. It is an open source web application, that provides an interactive environment to support the workflow of scientific computing from live code combination, execution, mathematics, plots, equations, visualizations, explanatory text to rich media[132]. We use this application to conduct the whole process of experiments, including data integrating, data cleaning, data preprocessing, data shifting, feature engineering, model building and statistical validating.

4.1.2 Machine Learning Toolkit

The Scikit-learn project in Python is used for the Machine Learning application. It is a strong and efficient library designed for diverse kind of users in terms of expertise. The dominant advantage of Scikit-learn is its user-friendly and easy-to-apply character. The primary aim of the project is to provide an econometric application with several classic, well-established machine learning regression algorithms, compare model forecasting performance and form an Ensemble Model. It includes tools for data pre-processing and feature engineering, machine learning model building and prediction, as well as tools

for model evaluation and selection. It provides efficient implementations of state-of-the-art algorithms, accessible to non-machine learning experts, and reusable across scientific disciplines and application fields.

4.2 Workflow of Experiment

The data used in this research is collected from multiple sources and is imported into Python Jupyter Notebook to conduct all the modelling. The analysis process consists of six steps listed below. Table 4.1 simplifies how the project operates. Figure 4.1 shows the work flow of the experiment design.

Table 4.1: Simplified Procedure Of The Project

Algorithm I: Simplified procedure of the project
1: Collect data from different sources and interpolate low frequency data
2: Conduct feature engineering and feature selection
3. Shift variables with period lag
4: Apply different machine learning algorithms and record their prediction results
5: Predictions from individual algorithm are combined with weighted voting
6: Compare the ensemble result with ARIMA and Random Walk models

1. The related predictors are chosen and an interpolation method is applied to the low frequency data to align with high frequency data to complete a balanced dataset.

2. Quarterly or yearly averaged and seasonally adjusted features are generated for each variable. Then, two features of each variable which are the most highly related to the target variable i.e. GDP quarterly percent change are selected for the modelling.

3. Features of different variables with period lag are shifted in proportion to their reporting delay to match up with the whole dataset.

4. Machine Learning algorithms such as Lightgbm, SVM, Ridge Regression, Lasso and Adaboosting are incorporated to do Nowcasting. The forecasting ability of these algorithms are compared with Mean Squared Error(MSE) and Median Absolute Error(MAE) value as evaluation. The ones with good model performance are selected into the next step while algorithms with rather weak forecasting abilities are abandoned.

5. An Ensemble Model, which is a weighted combination of GDP growth rate prediction results produced by the chosen approaches above, is created. The weight for each algorithm is based on its model forecasting abilities presented by the evaluation

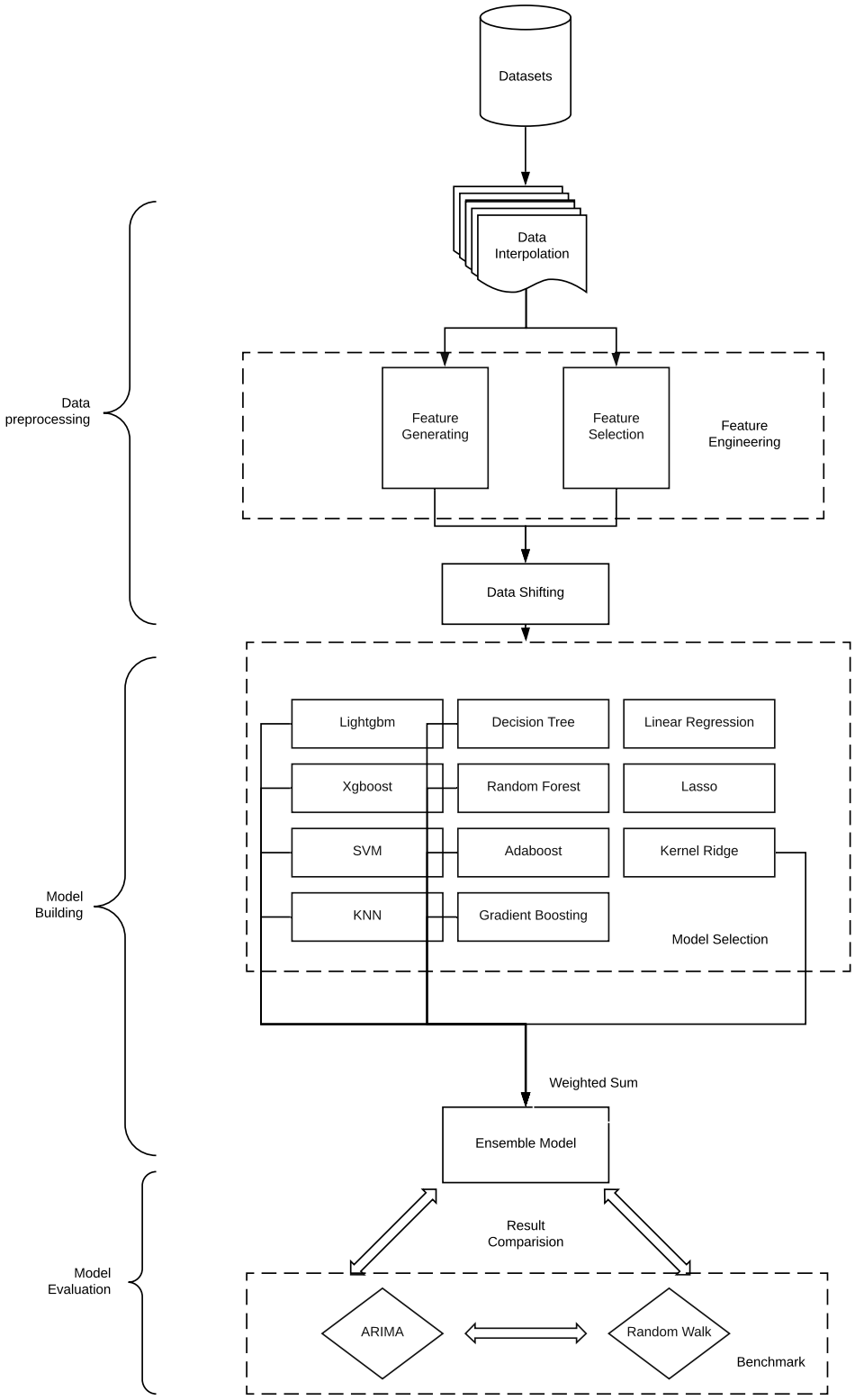


Figure 4.1: Work Flow of Experiment Design

indicator MSE. Through this process, the instability and prediction accuracy for the Ensemble Model is significantly improved.

6. The nowcasting result of the Ensemble model is compared with benchmark: ARIMA and Random Walk model in terms of MSE and MAE value.

4.3 Data

Variables from different data sources are integrated to form the data set. Table 4.2 and Table 4.3 provide the basic information regarding these variables. The target variable is GDP value in New Zealand which has been released by the government every quarter from 2000 to 2018. Input variables are a combination of hard and soft economical variables. Hard indicators include real economical data such as Price Index, Labor and Consumption, which are common components used for calculating GDP. Soft indicators cover Traffic Volumes on the state highways, which are key proxy variables of the economic situation in a country where freight relies mainly on land transportation like New Zealand.

Table 4.2: Variables For This Research

Variables	Unit	Source
Gross Domestic Product(GDP)	NZD	Stats NZ
Heavy Vehicles On State Highways(HV)-National Telemetry Sites sum	Per Vehicle	NZTA
Light Vehicle On State Highways(LV) -National Telemetry Sites sum	Per Vehicle	NZTA
Consumer Price Index(CPI)	Index	Reserve Bank
Population(PO)	Per Person	Reserve Bank
Unemployment Rate(UR)	Percent	Reserve Bank
Transaction Total Count(TTC)	Per Transaction	Paymark
Transaction Financial Amount(TFA)	Per NZD	Paymark
Terminal Count(TC)	Per Teminal	Paymark

Table 4.3: Brief Summary of Variable Statistics

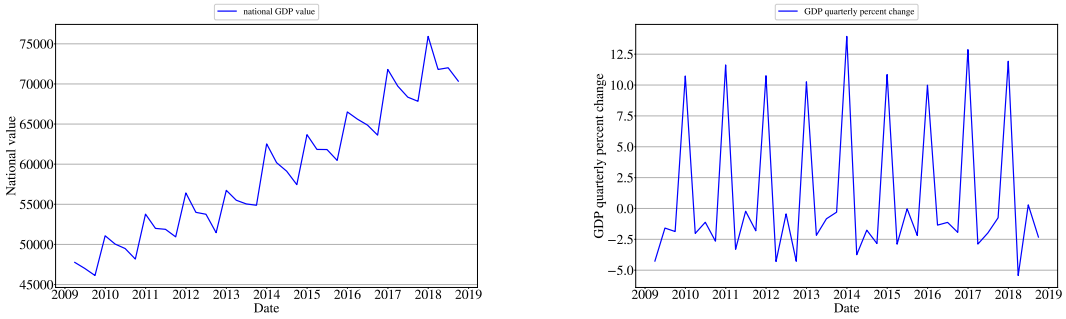
Variable	Start Date	Finish Date	Frequency	Count	Mean	SD.	Correlation with GDP
GDP	31/03/2000	31/12/2018	quarterly	75	49259.01	6590.93	1.00
HV	31/01/2000	31/12/2018	monthly	228	72115.25	25759.94	0.96
LV	31/01/2000	31/12/2018	monthly	228	769398.10	266652.00	0.94
CPI	31/03/2000	31/12/2018	quarterly	76	872.55	105.09	0.95
PO	31/03/2000	31/12/2018	quarterly	76	4323812.00	291991.40	0.99
UR	31/03/2000	31/09/2018	quarterly	75	5.05	0.91	-0.06
TTC	1/01/2009	15/01/2019	daily	3665	2483927.00	862971.20	0.94
TFA	1/01/2009	15/01/2019	daily	3665	2483927.00	862971.20	0.88
TC	1/01/2009	15/01/2019	daily	3665	55463.39	18012.20	0.82

* GDP: Gross Domestic Product HV: Heavy Vehicle LV: Light Vehicle
CPI: Consumer Price Index PO: Population UR: Unemployment Rate
TTC: Transaction Total Count TFA: Transaction Financial Amount TC: Terminal Count

Each variable has to be at least 40 quarters of continuous data available from 2009 to 2018 while the frequency of each variable varies from quarterly, monthly to daily.

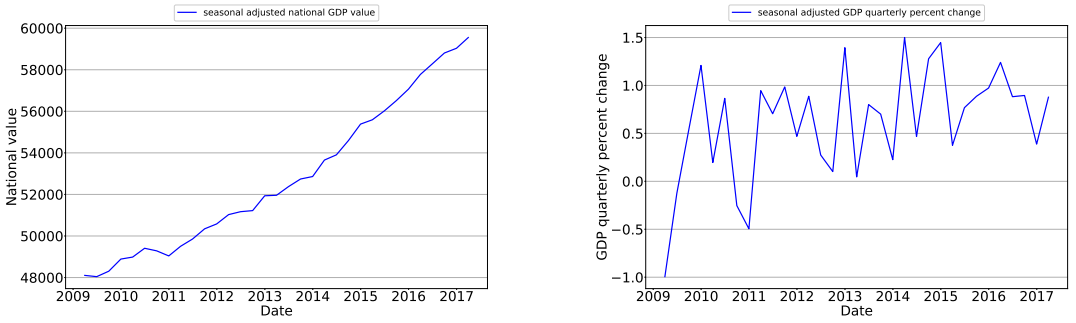
4.3.1 Target Variable Selection

GDP figures are acquired from the government statistics website Stats NZ on a quarterly basis. GDP data is made available approximately a quarter and therefore 20 days following the period it described. Figure 4.2 shows the basic trend of actual nominal New Zealand GDP as well as its quarterly percent change from 2000-03 to 2018-09 (the latest reporting prior to project completion).



(a) National Value (b) Quarterly Percent Change

Figure 4.2: Plot of GDP Figures



(a) National Value (b) Quarterly Percent Change

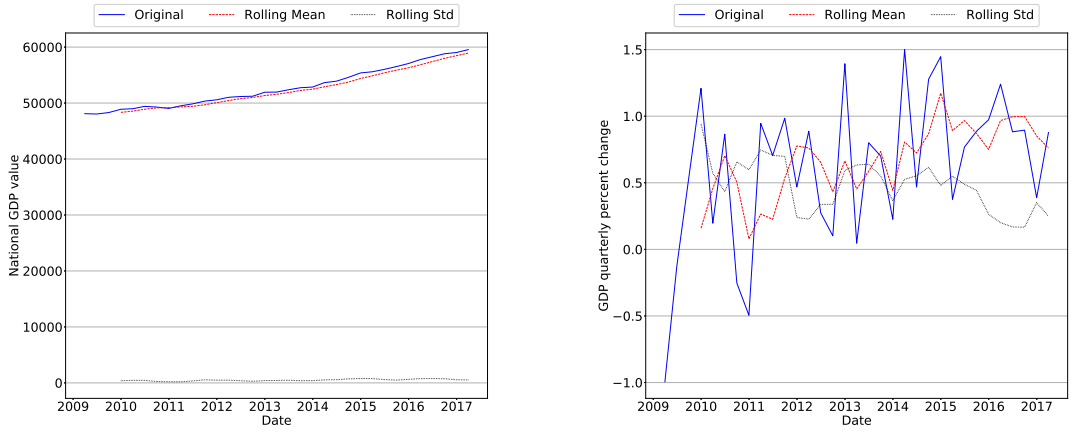
Figure 4.3: Plot of Seasonally Adjusted GDP Figures

As is common for economical statistics figures, the GDP of New Zealand has shown a

constantly increasing long-term trend and significant seasonality. Montgomery, Jennings, and Kulahci have pointed out in *Introduction to time series analysis and forecasting* : 'it is essential to recognize the seasonality and remove this component so that it will not mixed up with long-term trends [133].' This is known as seasonal adjustment.

To remove the trend and seasonality noise, GDP figures are chosen from the category "chain value, seasonal adjusted" category. Figure 4.3 illustrates the trend of both national GDP value and its quarterly percent change. Figure 4.3 shows the seasonal adjusted trend of both national GDP value and GDP quarterly percent change. The national value of GDP (Figure 4.2a) is more flat after eliminating the influence of seasonality (Figure 4.3a) while the quarterly percent change (Figure 4.2b) shows a more constant pattern in Figure Figure 4.3b since both of the increasing trend and seasonality of GDP has been removed.

Most of the time series data prediction models have relied on the strong assumption that the statistical properties of the data such as mean and variance should remain constant over time, i.e. the trend keeps stationary and is suitably mixed [134]. The prediction for a stationary variable would be regarded as accurate and reliable because that a higher possibility is believed for a stationary time series data to follow its steady historical pattern in the future.



(a) National GDP

(b) GDP Quarterly Percent Change

Figure 4.4: Dickey-Fuller Test

We use the rolling statistics plots along with Dickey-Fuller test results to check with the stationarity [134]. Figure 4.4a and Figure 4.4b provide the Dickey-Fuller test results for seasonal adjusted GDP figures and its quarterly percent change respectively. It is obvious

Table 4.4: Dickey-Fuller Test Results

	Statistics	P Value
National GDP Value	0.341301	0.979162
GDP Quarterly Percent Change	-6.3311E+00	2.9050E-08

from the plot that although removing the seasonal component, the national GDP figure remains the long-term increasing trend over time and the big p value of Dickery-Fuller test result shown in Table 4.4 confirms it is not stationary. On the other hand,the static rolling mean and rolling deviation as well as the small p value of Dickery-Fuller test result shows that the quarterly percent change of seasonal adjusted GDP is stationary.

Based on the discussion above, to remove the trendy and seasonal component, quarterly percent change of GDP seasonal adjusted figures is chosen as target variable. However, the ongoing revision of the New Zealand GDP chain value in the last quarter each year will be an issue to check the model prediction accuracy since the historical GDP figures would be revised differently. Until this report was completed, we would use GDP figures up to September 2018 which is the version published on 20th December, 2018 as the label to build and test models. In other words, the target variables are applied based on the official GDP figures released on 20th December, 2018 and the estimated quarterly GDP change would be compared with this version of governmental report. After the next quarter of data being released, we would update our target variable for future live prediction.

4.3.2 Input Data Selection

A rich and representative input dataset has been built combining up-to-date industry data with key macroeconomic data. Eight variables from different data source including Traffic Volume on the motor way, Consumer Price Index, Population, Unemployment Rate and the Household Consumption have been selected. Table 4.3 gives the brief statistical information of the eight input variables including Mean, Standard Deviation as well the correlation with national GDP figures.

Traffic Flow Variables

Traffic Flow is a real time and real world indicator for an economy, especially for a country like New Zealand, where a large proportion of freight movement is carried by road. Table 4.3 shows the strong correlation between GDP figures and heavy vehicle volumes (HV), light vehicle volumes (LV) on high ways. Large amount of vehicle numbers reflect more economic activities. Heavy Vehicles represents commercial activities while Light Vehicles

represent more private activities. Monthly heavy and light vehicle data from January 2000 to December 2018, covering 117 site stations are aggregated from New Zealand Transport Agency(NZTA) to predict national quarterly GDP growth.

Macroeconomic Indicators

Several public macroeconomic indicators from the Reserve Bank, which are highly correlated with GDP figures, are also employed in our model. They are Consumer Price Index, Unemployment Rate on a quarterly base from 2000-03 to 2019-12 and Population within the country on a quarterly base from 2000-03 to 2019-09. Higher level of Population and low level of Unemployment Rate increases the gross production ability. The relationship between CPI and economic growth is ambiguous under certain circumstances while in our case, it shows a positive correlation.

Data from Paymark

Household consumption is one of the largest components from the expenditure approach of GDP calculation. The key challenge for researchers in GDP prediction is the availability consumption data since information, such as credit card and financial transaction data, direct consumption information or communication data is often difficult to acquire due to privacy issues. Luckily, we have obtained secured Household consumption data provision from Paymark Ltd from 2009-01-01 to 2019-01-15. Paymark is a dominant company, which havs access to 75% of the market working on credit and debit/eftpost card transactions with 140,000 terminals in New Zealand.

The Paymark data is provided on daily basis, aggregated from different regions and industries across the country. The total number of transactions, the sum of financial transaction amounts as well as the number of terminals in operation is included. Since the nominal price level has been changing over time, the total transaction amount were adjusted by the most recent CPI figures provided by the Reserve Bank to reduce the influence of inflation. Meanwhile, to remove the effect of limited consumption ability caused by equipment, the sum of transaction numbers were normalized by terminal numbers at the corresponding time.

4.4 Data Interpolation

The start and finish dates from Table 4.3 suggests the unbalanced data frequency problem introduced by different data sources. To provide real time economic growth information for decision makers, the decision was made to do Nowcasting on a daily basis which could make the best use of high frequency input variables such as Paymark data without losing

useful information. Furthermore, daily level of granularity indicates a larger dataset with the same amount of features which would effectively solve the over-fitting problem caused by fat regression, i.e. various input data with limited observations.

Standard regression models require that all the input data should be in the same level of granularity. To align all the data properly with the higher frequency variable, interpolation is introduced to build a balanced dataset. The span of Paymark data, at present, is from 2009-01, all the other data were truncated and lined up with this coverage. Data points between each period of higher frequency variables are regarded as missing data and interpolated with spline interpolation method, which is widely used for non-linear data estimation. Dependent variable quarterly percent change on a daily basis are also generated using spline interpolation to align with the input dataset.

4.5 Feature Engineering

A raw dataset is like the crude oil for a Machine Learning algorithm from which proper features or predictors could be extracted. No matter how sophisticated the Machine Learning model is, without refining the raw dataset to meet the labels, accurate result could not be obtained. The process of building features for each variable while filtering the data used for feature based on the label's cutoff time to make valid features is called Feature Engineering. These features then will be passed as final input data to train the Machine Learning Model.

4.5.1 Feature Generating

An estimated quarterly percent change in CPI, as well as in Population are created for each point in time after interpolation. Meanwhile, the estimated Unemployment Rate on a daily basis is selected as a feature. To flatten the time series input data, a group of rolling calculations including rolling mean and rolling sum for one month, three month and six month, for each variable are derived as features. Similarly, features removing of seasonal influence as seasonal differences (the subtraction of a certain time point and the previous time point) and slope coefficients (the division of seasonal differences and corresponding time period) of one month, three months and six months are generated to guarantee the stationarity of input variables.

The transaction count variable are normalized by terminal counts to remove the effect of equipment over time while the financial value of total transportation is divided by the corresponding CPI to reduce the effect of inflation. In the end, the three months and six months rolling sum of Paymark data are selected as the representative features. Seasonal difference of a variable is measured by the difference between current value and the value

of previous year. For the transportation volume data and unemployment rate, the top ranked features correlated with GDP growth are the seasonal difference features. For variables like CPI and population, the seasonal trend matters as well as the quarterly percent growth.

4.5.2 Feature Selection

A correlation analysis is performed between features and respective dependent variables quarterly percent change of GDP. Two highest ranked features of each variable are selected from the correlation outcome as constituents of the total feature pool for machine learning model training. The final features fed into the models are listed in Table 4.5.

Table 4.5: Feature Summary

Feature	Explanation
HV_diff_year	Heavy Vehicle volume removing seasonal difference
HV_sum1_diff	One month rolling mean of HV removing seasonal difference
LV_diff_year	Light Vehicle volume removing seasonal difference
LV_sum1_diff	One month rolling mean of LV removing seasonal difference
CPI_diff_year	Consumer Price index removing seasonal difference
CPI_perchange	Estimated quarterly percent change of CPI
UR_diff_year	Unemployment rate removing seasonal difference
UR_sum1_diff	One month rolling mean of UR removing seasonal difference
PO_diff_year	Population removing seasonal difference
PO_perchange	Estimated quarterly percent change of Population
TRAN_COUNT_sum3	Three months rolling sum of transacation counts
TRAN_COUNT_sum6	Six months rolling sum of transacation counts
SUM_FIN_sum3	Three months rolling sum of total financial amount
SUM_FIN_sum6	Six months rolling sum of total financial amount
TERMINAL_sum3	Three months rolling sum of terminal counts
TERMINAL_sum6	Six months rolling sum of termnial counts

It is clear that for transportation variables like heavy vehicle and light vehicle volume, the differences between figures and one month rolling mean of figures at current time point and at the same time point in the previous year have the strongest correlation with GDP quarterly percent change. For macroeconomic data, features removing the seasonal difference as well as the estimated percent change outperform the others. As the Paymark consumption data, three months and six months rolling sum features are selected as the most relevant features.This is ascertained from the data structure and characteristics of each variable as well as its relationship with GDP quarterly percent change. The correlation between different features and their distribution are illustrated respectively in Figure 4.5 and Figure 4.6.

Although there is a strong relationship (above 0.5 or below -0.5) between each of the eight variables and the published GDP figures, the correlation between extracted features and the GDP growth is not that significant. As shown in the Figure 4.5, the highly

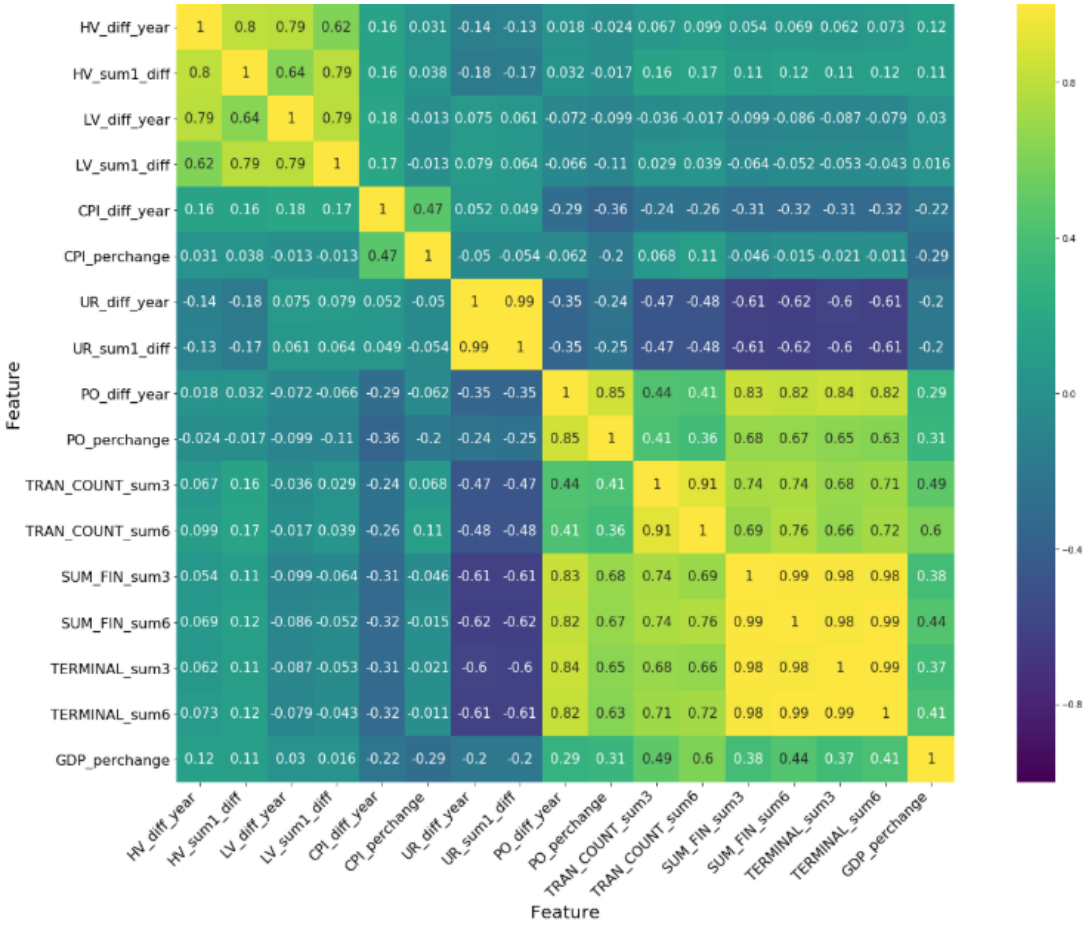


Figure 4.5: Correlation between Features

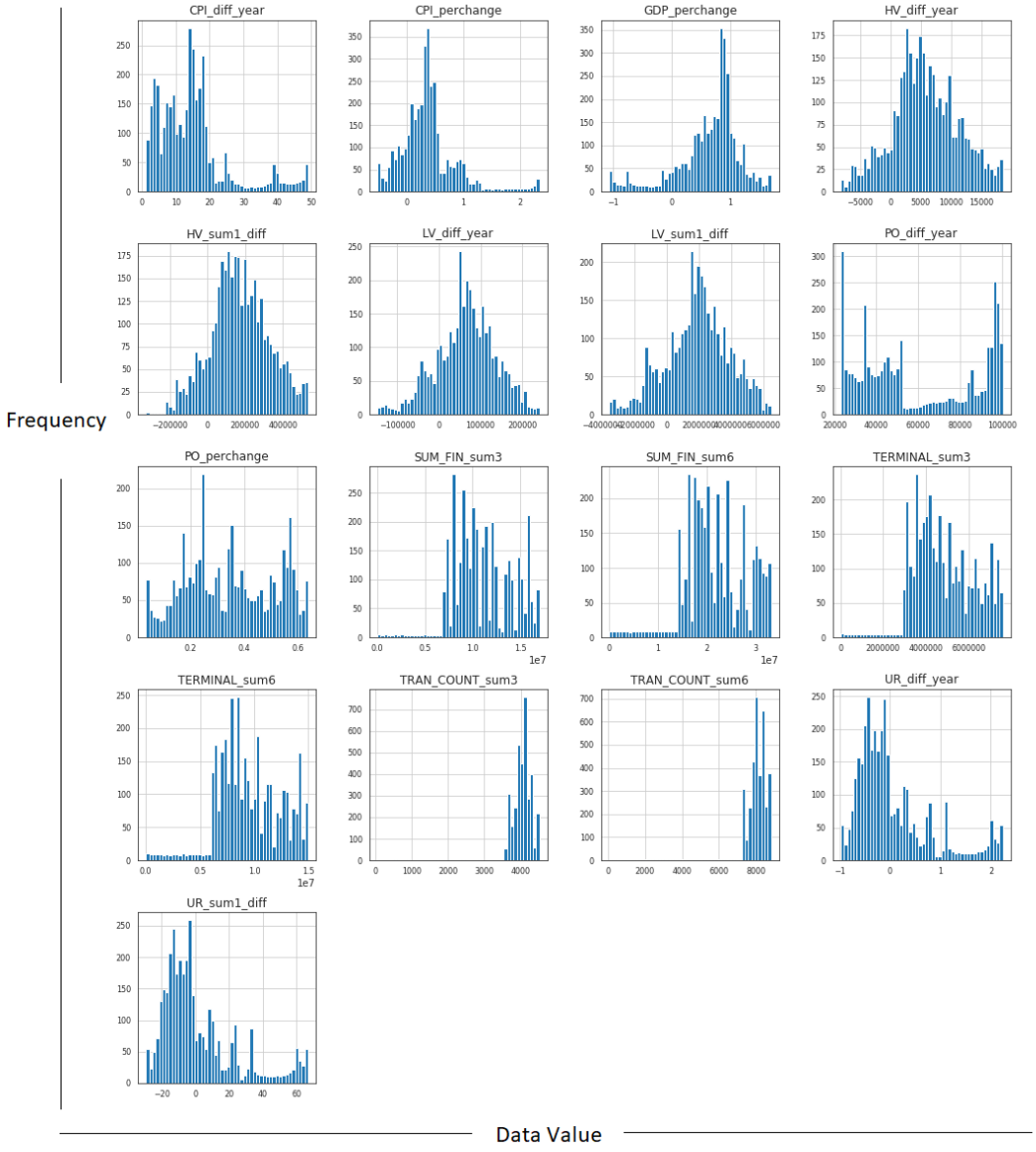


Figure 4.6: Distribution of All The Features

correlated features are from Paymark Data around 0.5, followed by macro-economical features around positive or negative 0.3. The relationship between transportation features and GDP percent change is less than 0.2. The relationship between GDP growth and each feature is consistent with the theory since private consumption reflected by Paymark Data is one of the elements calculating GDP while the macro-economical features indicating price, labor market and population have a direct influence on the economy. On the other hand, features of vehicle volume on state highways are just the proxy presenting the indirect relationship with GDP change.

The distribution of selected features as well as GDP percent change are presented in Figure 4.6. Heavy vehicles removing the seasonal trend and light vehicles without seasonal difference have shown the shape of a Gaussian distribution while the rest of features depict a long tail or heavy tail distribution. This highlights the fact that traffic volume features are more close to the random event showing a normal distribution while the rest of features are more centered to certain values.

4.6 Data Shifting

The concept of GDP real-time Nowcasting is to predict GDP figures at any given point in time. To satisfy this goal, a diverse resourced, timely fashion input dataset is required to make the actual predictions on an ongoing basis. However, the variables obtained from different sources have all different time span and release frequency, and besides, the reporting time of these data varies for a given period.

CPI (Consumer Price Index), Unemployment Rate have a reporting lag of two months, while Population has a delay of up to five months. This leads to the result that for those variables, only historical data with the time lag are available when it comes to the current time prediction. The strength of Machine Learning is that it could learn the pattern from the input variables itself. In this way, the historical data for the variables with lag is used to predict the current economic change. Once the data is interpolated into the daily basis, a shift of period is processed to their reporting delay to align with the current given time point. For example, to predict GDP quarterly growth on the date of 2019-01-15, the latest figure available for CPI is on 2018-12-31. A shift of 15 days would be carried to correct for this delay so that a complete input dataset with the same data span coverage could be fed into Machine Learning algorithm to make the in-time prediction.

Figure 4.7 shows the process of data shifting. Until all the experiments done, the latest data available in this research is on 2019-01-15 from Paymark, while the data from NZTA, CPI, unemployment dataset is up to 2018-12-31 and the population dataset available is up to 2018-09-30. To align with Paymark dataset, a 15-day-shift is conducted to NZTA

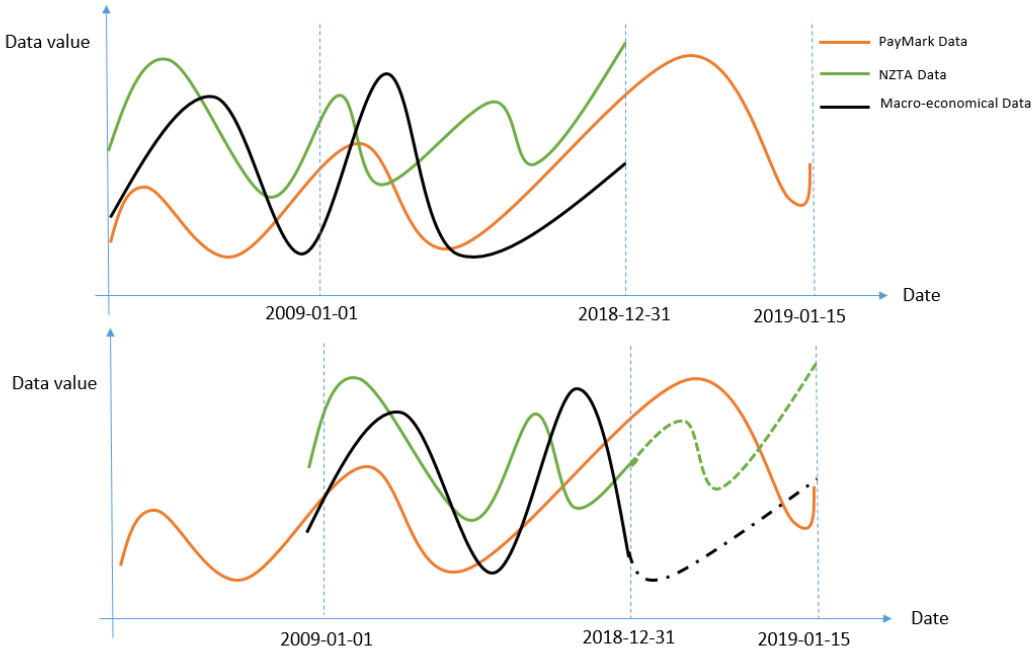


Figure 4.7: The Process of Data Shifting

dataset as well as CPI, unemployment data. A 105-day-shift is applied to Population data.

4.7 Forecast Evaluation Methodology

In this thesis, the most widely used econometric model for GDP Nowcasting model, ARIMA and Random Walk Model are used as the benchmark to compare with other Machine Learning Models. The Machine Learning Models' as well as the benchmark's performance to predict is conducted with an out-of-sample One-Step forecast exercise. A one-step forecast is a forecast of the very next time step in the sequence from the available data used to fit the model. This type of forecasting method is commonly used in Machine Learning domain.

Firstly, a comparison of recent quarter i.e 2018Q4 nowcasting performance of different machine learning algorithms is made by measuring the forecast accuracy of Mean Squared Error (MSE) and Median Absolute Error (MAE), defined as:

$$MSE(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{4.1}$$

$$MAE(y_i, \hat{y}_i) = median(|y_1 - \hat{y}_1|, ..., |y_n - \hat{y}_n|) \tag{4.2}$$

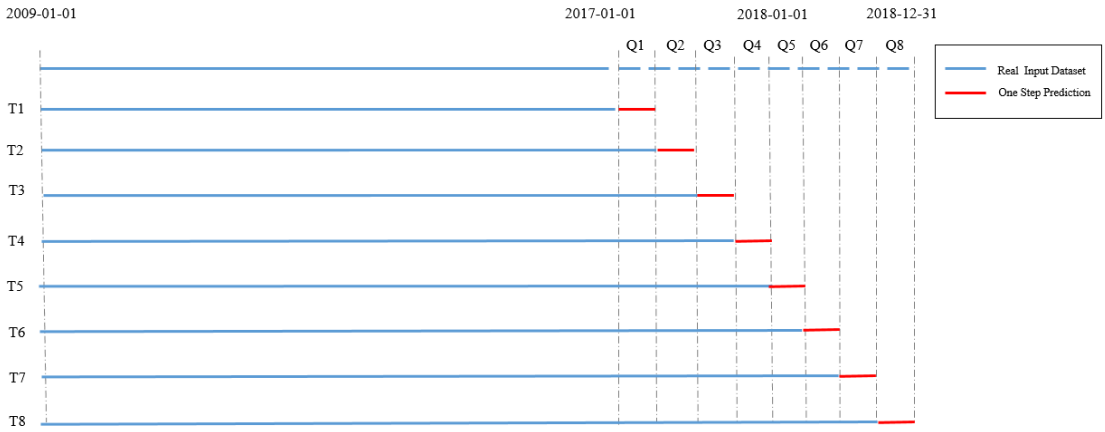


Figure 4.8: The Process of Expanding Window

where y_i stands for the actual values of GDP growth, $\hat{y}_i$ are the forecast values of GDP growth while $\bar{y}_i$ is the mean value of actual GDP change.

MSE and MAE are common evaluation metrics when performing regression. The Mean Squared Error measures the average of the squares of the difference between the estimated values and the actual value. The Median Absolute Error calculates the median of all absolute difference between the target and the prediction. It is robust to outliers. Other regression evaluation metrics such as R^2 score, Explained Variance Score are not valid for non-linear regression for the reason that their underlying assumption: Explained variance + Error variance = Total variance do not work in non-linear context. The choice of both MSE and MAE as evaluation methods guarantees the prediction accuracy without the influence of outliers.

The prediction results are combined into an Ensemble Method with the weighted voting based on each algorithm's forecast performance i.e MSE and we compare the prediction ability of ensemble method with each individual Machine Learning Algorithm.

Next, the robustness of our Ensemble Method is tested by training each Machine Learning algorithm over an expanding window thereby replicating an actual forecasting process starting from 2017Q1 and moving forward a quarter at a time through to 2018Q4. To be specific, for the first vintage of data, the models are estimated over the period 2009Q1 to 2016Q4 with real-time data for both predictor and response variables. The resultant fitted models are used to nowcast the growth of real GDP in next quarter which is 2017Q1. The process is shown in Figure 4.8. The ensemble prediction is calculated at each quarter by applying a weighting scheme to each individual algorithm's estimation. Overall, eight quarters of real time GDP Nowcasting results from 2017-01-01 to 2018-12-31 are generated and the models' prediction ability of eight quarters are measured as a whole.

The Forecast of ARIMA as well as the Random Walk Model as the benchmark are implemented using the same training/test split. ARIMA models are great for one-step forecasting. We use GDP growth on a quarterly base from 2008-12 to 2016-12 as training dataset and make the one-step (one quarter) prediction each time through to 2018-12. The eight steps (quarters) rolling forcasting are conducted with expanding the training dataset with next step prediction. The final ARIMA order (p,d,q) is selected by grid search which regarded the one with the smallest MSE through the eight quarters as the best model.

Chapter 5

Result Discussion

5.1 Out-of-Sample One-step Nowcasting

An input dataset, alligned with the Paymark variables from 2009-01-01 to 2018-12-31 has been built. Variables are integrated into this dataset including:

- **Two transportation variables:** Heavy Vehicles number and Light Vehicles number on state high ways from New Zealand Transport Agent (NZTA)
- **Three Macro-economical variables:** Consumer Price Index (CPI), Unemployment Rate and Population, from Reserve Bank
- **Three household consumption variables:** Transaction count, Transaction financial value and terminal count from PayMark

After feature engineering process to create features reflecting the rolling sum, rolling mean, seasonal difference and seasonal slope differences, two features of each variable are selected based on the correlation with GDP percent change, including:

- **Features for transportation variables:** HV_diff_year, HV_sum1_diff, LV_diff_year, LV_sum1_diff
- **Features for Macro-economical variables:** CPI_diff_year, CPI_perchange, UR_diff_year, UR_sum1_diff, PO_diff_year, PO_perchange
- **Features for household comsumption variables:** TRAN_COUNT_sum3, TRAN_COUNT_sum6, SUM_FIN_sum3, SUM_FIN_sum6, TERMINAL_sum3, TERMINAL_sum6

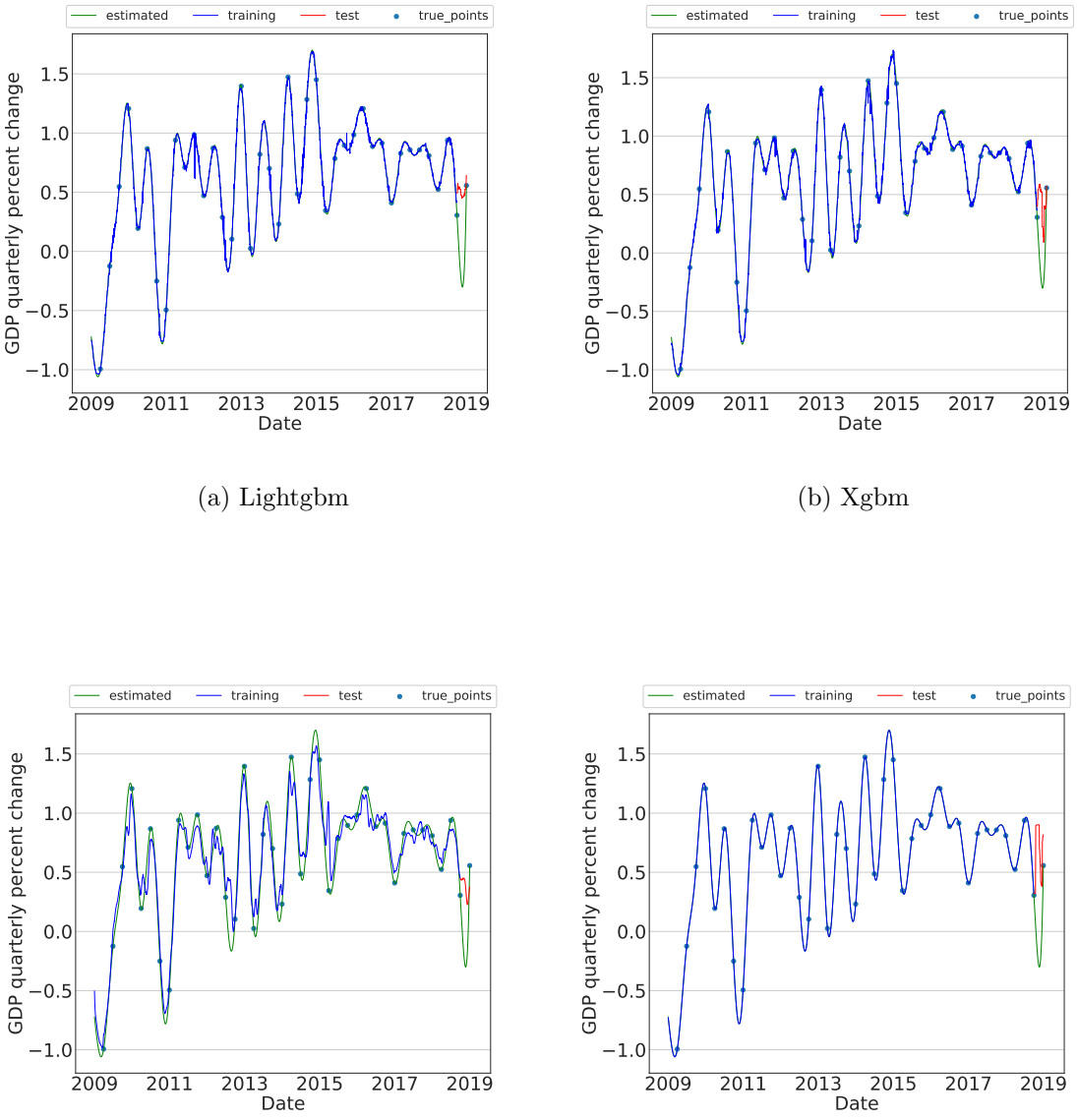
A shift of period is processed to their reporting delay to align with the current given time point.

It is common to fit a model with training data and then evaluate its performance on a test data set as out of sample data. When dealing with time series data, it is useful to compute one-step forecasts on the test data because the most recent step data remains part of the data structure and characteristic of the historical training data, the prediction result will be accurate. In this research, we have employed Out-of-Sample one-step Nowcasting. GDP figures are released quarterly so that one-step Nowcasting refers to predicting the most recent quarter of economic growth.

The dataset covering 10 years from 2009 to 2018 is split into training data (2009-01-01 to 2018-09-31) and last quarter testing data (2018-10-01 to 2018-12-31). A set of Machine Learning Methods are employed to make the one-step prediction, including Lightgbm, Xgboost, Support Vector Machine (SVM) with "rbf" kernel, K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Adaboost (AD), Gradientboost (GD), Lasso, Kernel Ridge (KR) regression with "laplacian" kernel and Linear Regression (LR). All these algorithms possess a range of tunable parameters. Grid search is used to select the best model with the MSE (mean squared error) as the evaluation method. Because of the special property of time series data, whose values are highly reliant on the time order, normal cross validation methods like K-Fold cross validation are not applicable. A Nested cross-validation loop, which rolls forward with several subsets is used to choose the best model with the right parameters.

The Ensemble Method is applied by combining the weighted prediction results of each individual machine learning algorithm with weight calculated from formula 3.19. Figure 5.1 presents the visualization results of the twelve algorithms, including the Ensemble Model.

As shown in Figure 5.1, models such as Lightbgm, Xgboost, Decision Tree and Random Forest perform well in detecting the trend of the training data while in the model variance part, those methods come across relatively serious over-fitting, i.e. less model prediction



(a) Lightgbm

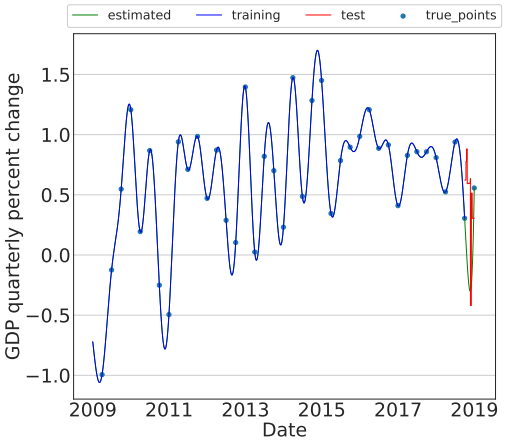
(b) Xgbm

(c) SVM

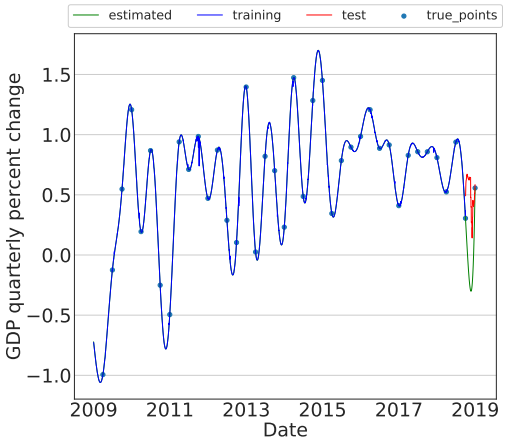
(d) KNN

Figure 5.1: Training and Test Prediction Results Comparison

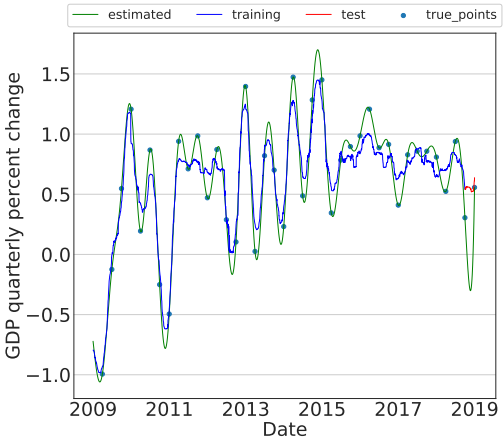
ability compared with training dataset. Lightbgm and Xgboost are proven to perform well especially with big data. The reason for serious over-fitting may be due to our limited data size.



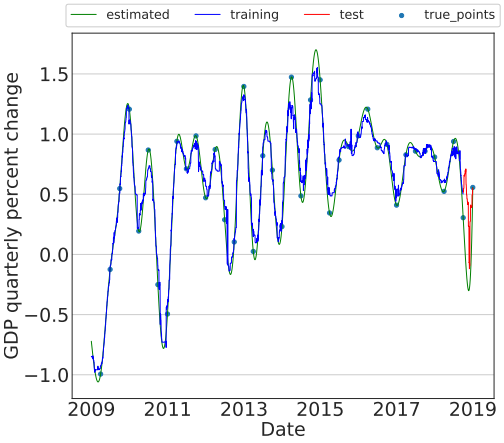
(e) Decision Tree



(f) Random Forest



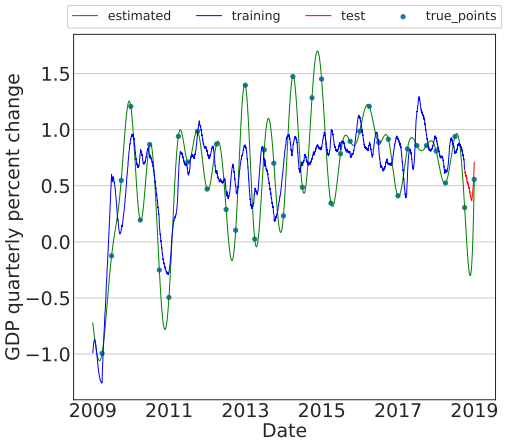
(g) Adaboost



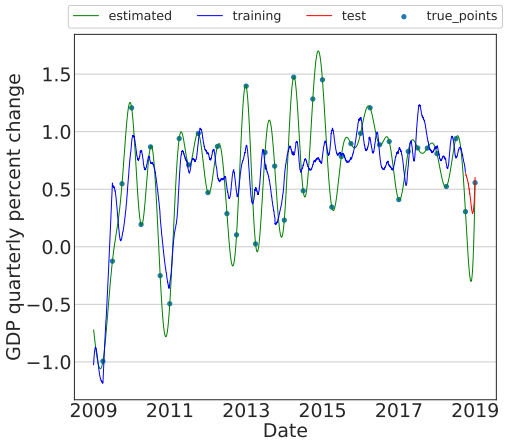
(h) GradientBoost

Figure 5.1: Training and Test Prediction Results Comparison

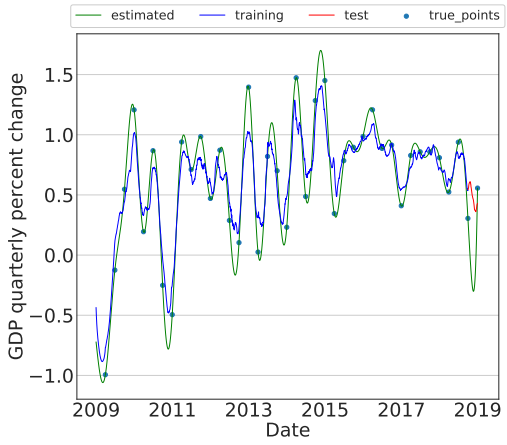
Adaboost and SVM algorithm present an improvement in model generalization resulting in the roughly fit approximation of the training dataset and a better prediction in the testing dataset. GradientBoost regression shows excellent accuracy as well as stable



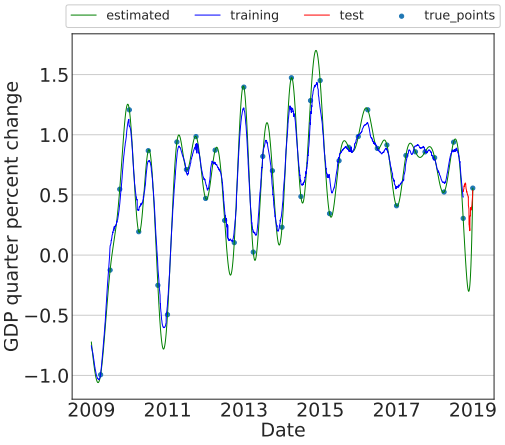
(i) Linear Regression



(j) Lasso



(k) Kernel Ridge



(l) Ensemble

Figure 5.1: Training and Test Prediction Results Comparison

generalization as noted in Figure 5.1h. However, compared to other regressors above, the performance of linear methods like Lasso and Linear Regression is not that satisfactory, the models capture the basic trend of GDP growth rather than fit the curve precisely while the Kernel Ridge algorithm, with non-linear kernel "laplacian" fits the training data better and generates a more accurate prediction result.

There are strengths and weaknesses in each algorithm as a trade-off of bias and variance for single model. Models like KNN and Decision Tree which reduce bias strictly turn out to have a limited model variance. Meanwhile, models like Lasso and Linear Regression have a good model generation ability but do not capture the subtle change in the training dataset. Among all the eleven algorithms, KNN presents the worst prediction ability that the estimated result for test data is in the opposite direction from the target data. In this way, the importance of KNN is excluded in the generating process of the Ensemble Method. The Ensemble Method in Figure 5.1l, combining all the algorithms other than KNN, could balance the bias and variance of each model and produce a relatively robust forecasting ability.

Table 5.1 and Table 5.2 illustrate the data evaluation matrix from the one-step prediction results for the methods above. The Mean Squared Error(MSE) as well as Median Absolute Error(MAE) for both training data and testing data are reported. Table 5.1 shows the MSE and MAE between the estimated GDP quarterly growth and the interpolated GDP quarterly percent change while table 5.2 provides the evaluation results for the real GDP quarterly growth i.e. 4 figures in a year including March, June, September, December. The weight of each algorithm to generate the Ensemble Method is calculated based on its prediction accuracy, following Formula 3.19. When considering the interpolated data evaluation, SVM, Gradientboosting, Xgboost and Lasso have gained the smallest MSE in the testing dataset so that they have bigger weights to construct the Ensemble Method while the importance of KNN is removed due to its worst model prediction ability.

MAE represents the loss between the target value and the predicted result by taking the median to remove the influence of outliers. Compared with MSE, the median absolute loss is generally bigger and consistent with MSE, the evaluation of MAE confirms again that the best performed models are SVM, Xgboost, Gradientboosting and Lasso. It is notable that almost all the models show a significant difference in both MSE and MAE between trainng data and testing data which implies weak model variance. However, the evaluation result in Table 5.1 measures the goodness of model fit to interpolated GDP change, those figures are generated via interpolation function as a possible reference of the real GDP growth, this results are satisfactory.

Table 5.2 gives the information of MSE and MAE value between the predicted GDP

Table 5.1: One-Step Prediction Evaluation: Interpolated Data Evaluation

Algorithm	Weight	Interpolated Data Evaluation			
		MSE_train	MSE_test	MAE_train	MAE_test
Lightgbm	0.0912	0.0005	0.3471	0.0087	0.6173
Xgboost	0.1169	0.0006	0.2708	0.0126	0.4928
SVM	0.1374	0.0181	0.2304	0.0928	0.4453
KNN	0.0000	0.0000	0.6358	0.0000	0.7628
DecisionTree	0.0800	0.0000	0.3957	0.0000	0.6816
RandomForest	0.0842	0.0000	0.3758	0.0010	0.5633
Adaboost	0.0779	0.0185	0.4064	0.1195	0.6495
GradientBoosting	0.1147	0.0076	0.2758	0.0473	0.5327
Lasso	0.1077	0.1150	0.2937	0.2193	0.5503
KernelRidge	0.0945	0.0284	0.3347	0.1175	0.6007
Linear	0.0955	0.1125	0.3313	0.2283	0.5768
Ensemble		0.0129	0.3035	0.0776	0.5367

Table 5.2: One-Step Prediction Evaluation: Real Data Evaluation

Algorithm	Weight	Real Data Evaluation			
		MSE_train	MSE_test	MAE_train	MAE_test
Lightgbm	0.0912	0.0007	0.0074	0.0086	0.0860
Xgboost	0.1169	0.0015	0.0002	0.0136	0.0151
SVM	0.1374	0.0233	0.0318	0.0924	0.1784
KNN	0.0000	0.0000	0.0681	0.0000	0.2609
DecisionTree	0.0800	0.0000	0.0634	0.0000	0.2517
RandomForest	0.0842	0.0002	0.0008	0.0009	0.0281
Adaboost	0.0779	0.0237	0.0066	0.1400	0.0811
GradientBoosting	0.1147	0.0109	0.0000	0.0681	0.0050
Lasso	0.1077	0.1341	0.0021	0.2680	0.0458
KernelRidge	0.0945	0.0349	0.0162	0.1427	0.1272
Linear	0.0955	0.1321	0.0237	0.2913	0.1540
Ensemble		0.0168	0.0004	0.1037	0.0193

growth and the official released GDP percent change by removing the interpolated data. There is only one of real GDP quarterly change data value in the test period which is on 2018-12-31. In this way, the MSE value is just the squared MAE value. The Gradientboosting method performs extremely well with a very small MSE and MAE value of test dataset indicating a very close prediction result with real GDP growth, followed by the Xgboost and Ensemble model.

Compared with the result of the estimated GDP growth, both the MSE and the MAE values of the test data for all the algorithms have reduced dramatically and the difference between the training data and the test data is not as big as for the interpolated evaluation result. This implies that our Nowcasting result is more accurate and reliable when considering the real GDP changes.

Previous researches have proved that the Ensemble Method could improve the prediction accuracy effectively by its function to incorporate each single model's unique feature for different data pattern and balance the limitation of single method for complex data structure. It can be seen from both Table 5.1 and Table 5.2 that the Ensemble Method has won the relatively superior model prediction performance. Considering the MSE value for interpolated training data, following Gradientboosting, Xgboost, SVM and Lasso, it has beaten the rest seven algorithms and gained the fifth place.n However, =the process of the single quarter One-step Nowcasting could only prove the effectiveness of the Ensemble Method to get a stable above average prediction model. To test the robustness and generalization of the Ensemble Method in different time period, an Expanding Window Validation strategy has been applied.

5.2 Expanding Window Validation

We split the 10 years, 40 quarters dataset into 32 quarters training data and 8 quarters Out-of-Sample testing dataset. Each algorithm is trained based on the period 2009Q1 to 2016Q4 and conducted the One-Step Nowcasting of 2017Q1. After this, the actual forecasting situation starting from 2017Q1 and moving forward a quarter at a time until 2018Q4 is replicated. The MSE values of each algorithm for the 8 quarter are recorded to generate the prediction results of the Ensemble Method. The nine algorithms including Lightgbm, Xgboost, Support Vector Machine, K-nearest Neighbours, Decision Tree, Random Forest, Adaboost, GradientBoosting and Kernel Ridge are combined to generate the best-fit Ensemble Method. Lasso and Linear Regression do not perform well in the non-linear regression situation and are abandoned in this session.

The MSE of all the algorithms along the process are shown in Table 5.3. From Table 5.3, it is clear that the Ensemble Method is more stable and remains the top four algo-

rithms with the smallest MSE value while No.1 ranking algorithm with the smallest MSE value varies during the 8 quarters. This proves the reliability and robustness of Ensemble Method.

Table 5.3: MSE values of All the Algorithms In The Expanding Window

Algorithm	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Lgbm	0.014	0.027	0.001	0.004	0.056	0.010	0.049	0.347
XGBM	0.016	0.011	0.002	0.003	0.118	0.015	0.039	0.271
SVM	0.033	0.157	0.007	0.016	0.050	0.012	0.019	0.230
KNN	0.034	0.072	0.015	0.001	0.064	0.010	0.049	0.636
DT	0.022	0.038	0.001	0.002	0.009	0.243	0.008	0.472
RF	0.006	0.015	0.005	0.013	0.154	0.040	0.021	0.372
AD	0.048	0.005	0.014	0.007	0.169	0.028	0.040	0.433
GB	0.025	0.009	0.002	0.001	0.162	0.018	0.036	0.218
KR	0.032	0.109	0.005	0.020	0.025	0.009	0.031	0.335
Ensemble	0.012	0.009	0.001	0.001	0.023	0.008	0.017	0.307

* Quarter 1:2017.01.01-2017.03.31, Quarter 2:2017.04.01-2017.06.31
Quarter 3:2017.07.01-2017.09.30, Quarter 4:2017.10.01-2017.12.31
Quarter 5:2018.01.01-2018.03.31, Quarter 6:2018.04.01-2018.06.31
Quarter 7:2018.07.01-2018.09.30, Quarter 8:2018.10.01-2018.12.31

Table 5.4: Algorithm Ranking in the Expanding Window

Quarter	Lgbm	Xgbm	SVM	KNN	DT	RF	AD	GB	KR	Ensemble
Q1	3	4	8	9	5	1	10	6	7	2
Q2	6	4	10	8	7	5	1	2	9	3
Q3	1	5	8	10	3	6	9	4	7	2
Q4	6	5	9	2	4	8	7	3	10	1
Q5	5	7	4	6	1	8	10	9	3	2
Q6	3	6	5	4	9	10	8	7	2	1
Q7	9	7	3	10	1	4	8	6	5	2
Q8	6	3	2	10	9	7	8	1	5	4
Total Rank	39	41	49	59	39	49	61	38	48	17
Average Rank	4.88	5.13	6.13	7.38	4.88	6.13	7.63	4.75	6	2.125
Friedman Test: Statistics= 19.145, p=0.024 < 0.05										

The algorithm ranking according to their MSE value has been shown in Table 5.4. Since a smaller MSE value suggests the higher model forecasting ability, model with the smallest MSE gets No.1 ranking in this process. Total Rank which is the sum of each model's ranking along the 8 quarters, as well as the Average Rank along this process is calculated. The ranking of different algorithms varies from quarter to quarter while the Ensemble method performs stable in terms of model forecasting ability and gains the smallest ranking in both total rank and average rank.

A statistical test, Friedman test is applied with the algorithm ranking dataset shown in Table 5.4 to test whether there is a significant difference in the forecasting ability among all the models. The null hypothesis for the Friedman test is that there are no differences between the variables, if the p value is less than the significance level, the null hypothesis is

rejected and it could be concluded that at least 2 of the variables are significantly different from each other. A significance level of 0.05 is used in this research. The calculated p value 0.024 is less than 0.05 which confirms the Ensemble Method is more accurate significantly than the others in a statistical level.

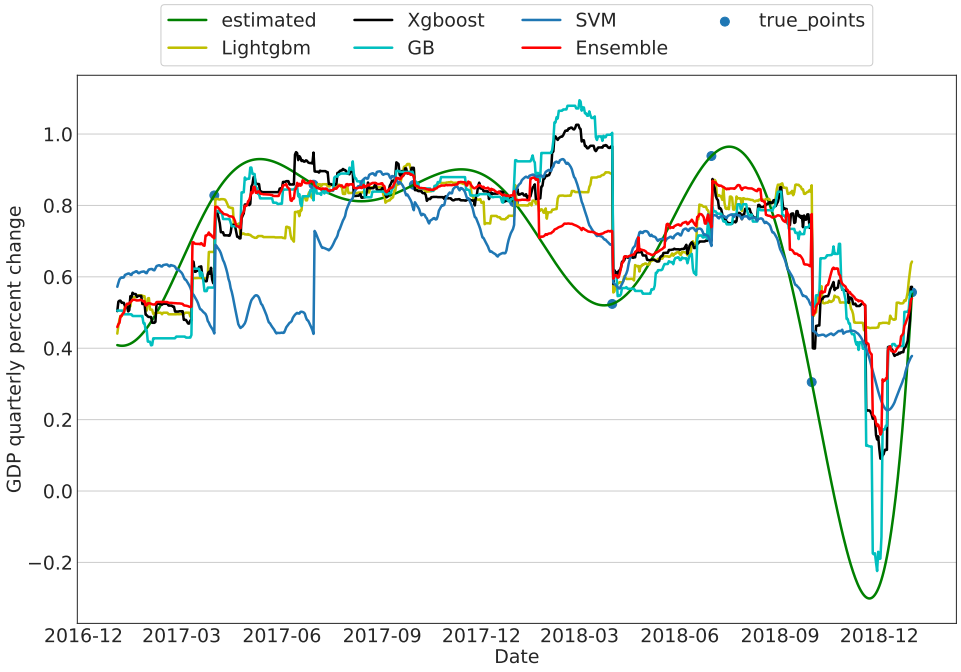


Figure 5.2: Prediction Results Comparison for Lightgbm, Xgboost, Gradient Boosting, SVM and the Ensemble Method from 2017-01-01 to 2018-12-31

The prediction results covering all the 8 quarters are evaluated, as a whole, with MSE and MAE score. The visualization results of the ten algorithms including Ensemble Method are shown in Figure 5.2 and Figure 5.3.In Figure 5.2, GradientBoosting as well as Xgboost, Lightgbm shows a steep fluctuation which do not closely follow the tendency of how GDP changes. What is interesting in Figure 5.3 is that algorithms like KNN and decision tree which make the prediction based on the points nearby present a deeply jagged trend. Of all the 10 algorithms, Ensemble Method neutralizes all the weakness of the individual model and combines all the advantages. The Ensemble Model displays a stable forecasting ability that stimulates closely to the estimated GDP quarterly change.

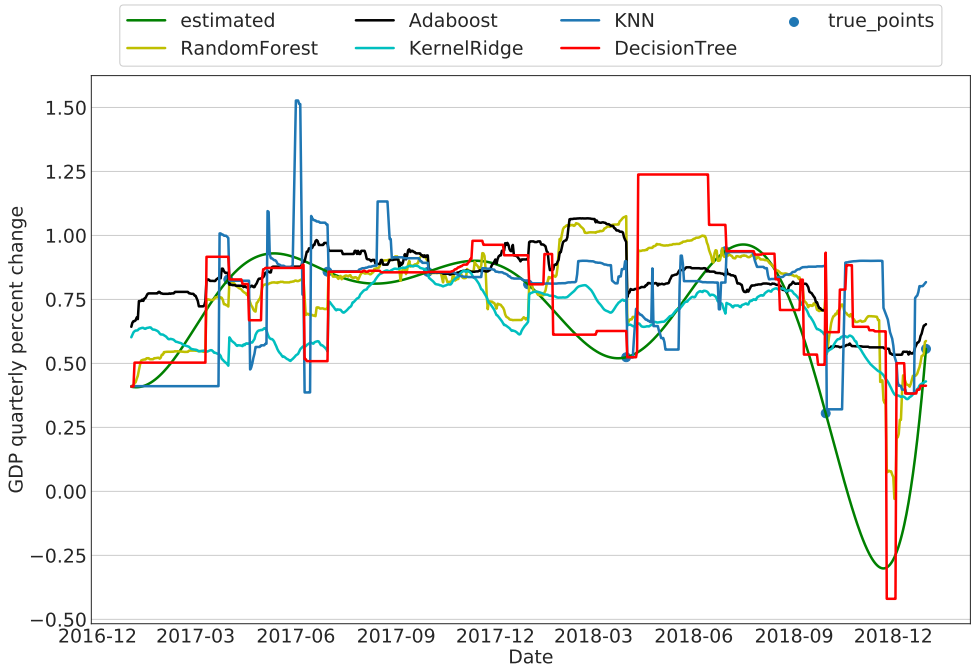


Figure 5.3: Prediction Results Comparison for Decision Tree, Random Forest, Kernel Ridge, KNN and Adaboost Method from 2017-01-01 to 2018-12-31

The evaluation of the prediction abilities of all 10 algorithms from 2017-01-01 to 2018-12-31 is given in Table 5.8. Table 5.8 shows that Ensemble Method outperforms all the rest individual algorithms both in interpolated data evaluation and real data evaluation. It has the smallest MSE of 0.047 considering the interpolated GDP growth and the MSE value even decreased to 0.039 after removing the estimated GDP quarterly change. Most of the individual algorithm's MSE increased along this process. The MAE value of Ensemble method reaches to the lowest 0.083 in interpolated data evaluation, similar with KNN. The second lowest MAE value is from Xgboost 0.092. In the real data evaluation, Ensemble method wins the smallest MAE value 0.111 while the second smallest MAE value comes from Random Forest 0.116.

5.3 Comparison with Random Walk and ARIMA models

The benchmark ARIMA and Random Walk model are estimated on a quarterly basis with real GDP growth as well as the Random Walk model which makes a prediction only according to the value of previous time point. For a fair comparison with Ensemble method, training data from 2009-01 to 2016-12 is imported into ARIMA and Random Walk model, One-Step Nowcasting is made along the following 8 quarters up to 2018-12. The parameters (6,2,2) of ARIMA model generating the smallest MSE value is selected by grid search. Table 5.5 shows the prediction evaluation of real GDP quarter change data covering 8 quarters.

Table 5.5: MSE value for the Real Data Evaluation in The Expanding Window

Algorithm	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Lgbm	0.025	0.001	0.002	0.001	0.132	0.048	0.304	0.007
Xgbm	0.061	0.008	0.002	0.000	0.195	0.056	0.196	0.000
SVM	0.150	0.175	0.005	0.004	0.027	0.064	0.048	0.032
KNN	0.026	0.034	0.001	0.000	0.141	0.015	0.333	0.068
DT	0.081	0.105	0.000	0.013	0.008	0.011	0.276	0.025
RF	0.008	0.005	0.001	0.016	0.329	0.005	0.255	0.000
AD	0.000	0.001	0.001	0.028	0.166	0.004	0.229	0.008
GB	0.079	0.001	0.000	0.003	0.245	0.054	0.195	0.000
KR	0.114	0.097	0.000	0.021	0.048	0.060	0.096	0.016
Ensemble	0.028	0.000	0.001	0.001	0.042	0.039	0.203	0.000
Randomwalk	0.139	0.001	0.003	0.011	0.113	0.139	0.470	0.047
ARIMA	0.007	0.023	0.012	0.167	0.002	0.093	0.013	0.004

Compared with the evaluation results of estimated GDP growth in Table 5.3, the MSE value of the best model in each quarter has decreased, the MSE value of some models in a specific quarter even reaches 0.000. This confirms that Machine Learning Models are more accurate taking only real GDP quarterly growth into account. It can be seen from Table 5.5, ARIMA model outperforms Randomwalk as well as other models in general and is close to Ensemble Model in terms of forecasting ability.

Table 5.6 summarises the ranking of 10 models as well as 2 benchmarks according to the MSE value along 8 quarters. Among all the methods, Ensemble model has the highest ranking in both total rank and average rank while the Friedman Test shows there is not a significant difference among all the methods.

Table 5.6: Algorithm Ranking for the Real Data Evaluation in The Expanding Window

Quarter	Lgbm	Xgbm	SVM	KNN	DT	RF	AD	GB	KR	Ensemble	ARIMA	Randomwalk
Q1	4	7	12	5	9	3	1	8	10	6	2	11
Q2	5	7	12	9	11	6	3	4	10	1	8	2
Q3	8	9	11	6	2	7	5	3	1	4	12	10
Q4	4	2	6	1	8	9	11	5	10	3	12	7
Q5	7	10	3	8	2	12	9	11	5	4	1	6
Q6	6	8	10	4	3	2	1	7	9	5	11	12
Q7	10	5	2	11	9	8	7	4	3	6	1	12
Q8	6	2	10	12	9	4	7	1	8	3	5	11
Total Rank	50	50	66	56	53	51	44	43	56	32	52	71
Average Rank	11.1	11.1	14.7	12.4	11.8	11.3	9.8	9.6	12.4	7.1	11.6	15.8
Friedman Test: Statistics= 11.0, p=0.443 > 0.05												

Table 5.7: Wilcoxon Signed-Rank Test

	Statistics	P Value
(Ensemble,ARIMA)	9.00	0.208
(Ensemble,RandomWalk)	0.00	0.012

To compare the forecasting performance between Ensemble method and the two benchmarks in detail, a Wilcoxon Signed-Rank Test is undertaken. Wilcoxon Signed-Rank Test is to test whether or not the distribution of two paired samples are equal. The result in Table **??** shows that Ensemble method outperforms Random Walk model significantly while there is not remarkable difference in model prediction ability between The Ensemble Model and ARIMA.

Table 5.8: Prediction Results Evaluation from 2017-01-01 to 2018-12-31

	Interpolated Data Evaluation		Real Data Evaluation	
Algorithm	MSE	MAE	MSE_real	MAE_real
Lgbm	0.063	0.106	0.065	0.122
XGBM	0.059	0.092	0.065	0.163
SVM	0.066	0.160	0.063	0.198
KNN	0.110	0.083	0.077	0.173
DT	0.099	0.079	0.065	0.233
RF	0.078	0.125	0.077	0.116
AD	0.093	0.111	0.055	0.096
GB	0.059	0.111	0.072	0.148
KR	0.070	0.159	0.056	0.231
Ensemble	0.047	0.083	0.039	0.111
RandomWalk			0.115	0.277
ARIMA(6,2,2)			0.040	0.113

Table 5.8 provides the information of the overall model performance covering the 8 quarters. Compared with all the machine learning algorithms, Random Walk model does not fit the real GDP growth from 2017-01 to 2018-12 well given the relatively large MSE value 0.115 while ARIMA outperforms all the individual model with a small MSE value of 0.040. However, the Ensemble method wins a smaller MSE value of 0.039 and a smaller MAE score of 0.111 (the MAE avlue of ARIMA is 0.113). This again confirms the statistical result that Ensemble Model is significantly superior than the Random Walk model and slightly outperforms the ARIMA model.

The Prediction visualization of ARIMA, Random Walk and Ensemble Method is given by Figure 5.4. The forecasting results shown in Figure 5.4 are consistent with the evaluation. Random model has not fit well with the real GDP growth, ARIMA follows the basic trend of the real GDP quarterly change, while the Ensemble method almost captures all the subtle change of the economic growth. This is another strong proof of the excellent model prediction ability as well as the robustness of Ensemble Method.

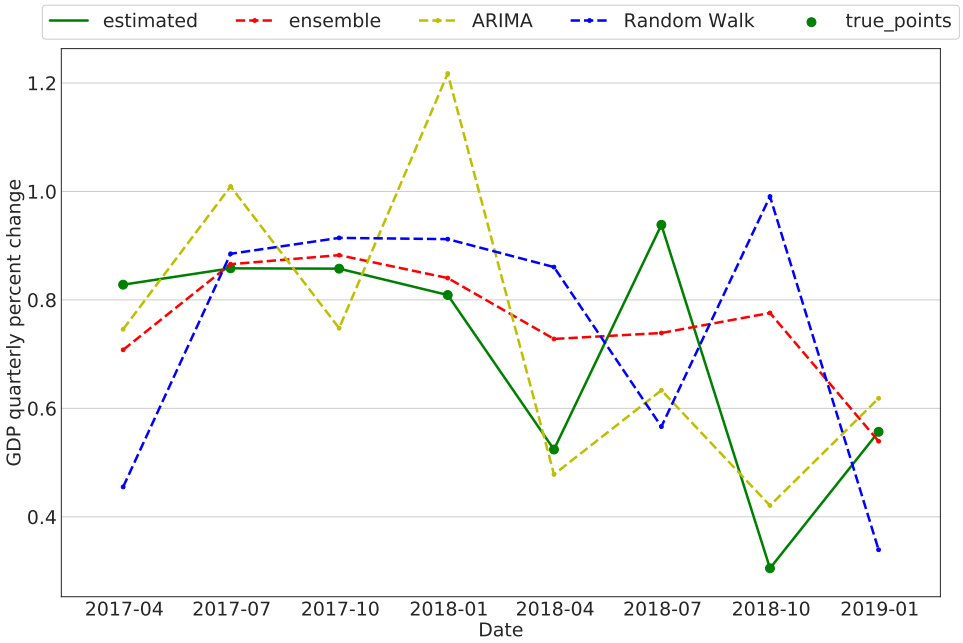


Figure 5.4: Prediction Results of Ensemble Method, ARIMA and Random Walk from 2017-01-01 to 2018-12-31 Visualization

Although,there is not enough evident to show the superiority of the Ensemble Model upon ARIMA model. As discussed in Chapter 2, univariate autoregressive models like ARIMA and Random Walk model predict time series data only based on its historic data vintage. As for economic situation approximation, univariate autoregressive models could work well in the stable economical situation but it lacks the foresight to predict the hidden economical change. The Ensemble Model mapping input variables extracted from rich high-frequency timely variables to GDP growth performs more reliably in both stable and fluctuating situations.

Chapter 6

Conclusions

6.1 Discussions

The initial idea of this study is to create a live GDP predictor in New Zealand reflecting current economic situation in the period of government statistics release delay. An Machine Learning approach instead of traditional econometric approach is taken due to the adjustment parameters to minimize the predicting error of the learning algorithm and its remarkable forecasting ability for non-linear data.

In this research, the real-time performances of eleven popular Machine Learning Algorithms are evaluated in Nowcasting the Gross Domestic Product (GDP) quarterly change for New Zealand. Different machine learning models are applied over 2009-2018 period a large number of features extracted from macro-economic variables, traffic proxy and customer transactions information. The prediction results of the nine out of eleven Machine Learning Algorithms are combined to generate an Ensemble Model by weighting them based on their forecasting accuracy. Lastly the prediction ability of the Ensemble Model is compared with naive autoregressive benchmarks ARIMA model and Random Walk Model, as well as each individual Machine Learning Algorithm.

The results demonstrate that the application of the Ensemble Model constructed by nine Machine Learning methods outperformed traditional economical approach for GDP forecasting as well as each single Machine Learning algorithm in terms of both prediction accuracy and model reliability. Using data from 2009Q1 to 2016Q4 as training data set and the Expanding Window strategy, the Mean Squared Error of the final ensemble model for testing data set (2017Q1 to 2018Q4) is 0.039 while the benchmark ARIMA model is 0.040. This provides strong evidence that real-time GDP Nowcasting at the level of daily granularity is achievable. Our work has filled the research gap of GDP Nowcasting in New Zealand and offered relatively timely and accurate economical trend prediction results for decision makers.

The novelty of this research lies in the following three aspects: Firstly, it integrates a large set of input dataset from multiple data sources such as NZTA, Stats NZ, Paymark and Reserve Bank. Thanks to our honorable cooperators, this dataset, which including public macroeconomic variables, proxy indicators as well as individual consumption data, is the first and most comprehensive input dataset for GDP prediction in New Zealand.

Secondly, it presents the application experiences of eleven Machine Learning algorithms for GDP prediction in New Zealand. Limited by the unvanquishable computational difficulties and the small size of macroeconomic indicators, the traditional Econometric Approach has been dominated GDP Nowcasting area with linear statistical methods. However, in the recent two decades, data-driven approach like Machine Learning has shown a superior prediction performance especially with a great number and diversity of data in a non-linear data structure. With the assistant of high frequency variables from different data sources and the interpolation strategy to handle the unbalanced data problem, we have tested eleven Machine Learning Algorithms and recorded their prediction accuracy.

Lastly, it creates an Ensemble Model which combines all the Machine Learning Models with a system of averaging method. Empirical evidences have shown that the accuracy and the reliability of a forecasting model could be improved significantly by combine multiple individual models together. By taking the weights based on each single Machine Learning Algorithm's prediction accuracy, an Ensemble Model is generated and has proved to be superior significantly than each single method and the benchmarks (ARIMA and Random Walk Model).It is not surprising that at this stage the accuracy of predictions has some room for improving. Considering the complex of social and economical phenomenon as GDP itself, as well as the various parameters and strategies in Machine Learning area, the prediction of GDP will be accompanied with uncertainty and potential errors.

6.2 Limitations and Delimitations

There are two aspects of limitations associated with our exploratory research.

1. The data source limitation. The motivation of our work is to make predictions on economic activities based on the rich and large volumes of high frequency data while the current model only covers dataset from transportation, consumption, population and job market. The data with information of the production side such as the Primary Industries has not been included yet. Besides,the real daily dataset we involved is only from Paymark, the interpolation of monthly and quarterly dataset will bring in unwanted bias with the use of the spline interpolation injects which estimates interval data points by the approaching the next known GDP points.

2. The model selection limitation. In light of the various parameter tuning and strategy

selecting in machine learning area, it is never confirmed that the best model has been employed. There are machine learning models as well as ensemble methods which may generate more accurate results. Recently, recurrent networks in the deep learning area especially the long and short term memory networks with their extraordinary model ability to handle the memory of historical vintage has gain remarkable performance in the forecasting of time series. However, limited by the small size of current dataset, the complete dataset including GDP quarterly change and all the regressors from 2010Q1 to 20189Q4, the implementation of deep learning is not effective. This could be addressed by either enriching the current dataset or generating more possible estimated data points. Furthermore, we have not covered the comparison of machine learning models with the traditional statistical models such as Bridge Equation and Spike-and-Slab Regression, which could be the focus for the future research.

6.3 Future Work

Future efforts could focus on addressing the limitations highlighted above:

1. Integration of more high frequency real time data from KiwiRail, Portconnect and NZTA data source as well as the Primary Indestries Sector will be conducted. In addition, multiple version of the predict variables will be generated and modelled.

2. Model parameter adjustments should continue to be fine-tuned and a more comprehensive Ensemble Modelling strategy considering the decaying effect of previous data should be introduced into the model averaging process.

Reference

[1] P. A. Samuelson, *Economics-an introductory analysis*. Mcgraw-Hill Book Company, Inc.; New York; London; Toronto, 1951.

[2] D. Giannone, L. Reichlin, and D. Small, "Nowcasting: The real-time informational content of macroeconomic data," *Journal of Monetary Economics*, vol. 55, no. 4, pp. 665–676, 2008.

[3] M. Bańbura, D. Giannone, M. Modugno, and L. Reichlin, "Now-casting and the real-time data flow," in *Handbook of economic forecasting*, vol. 2, Elsevier, 2013, pp. 195–237.

[4] A. Richardson, T. Mulder, *et al.*, "Nowcasting new zealand gdp using machine learning algorithms," 2018.

[5] N. J. Nilsson, *Introduction to machine learning: An early draft of a proposed textbook*, 1996.

[6] J. VanderPlas, *Python data science handbook: Essential tools for working with data.* " O'Reilly Media, Inc.", 2016.

[7] J. Luan, "Data mining and its applications in higher education," *New directions for institutional research*, vol. 2002, no. 113, pp. 17–36, 2002.

[8] C. Nyce and A. CPCU, "Predictive analytics white paper," *American Institute for CPCU. Insurance Institute of America*, pp. 9–10, 2007.

[9] J. Duan, "Nowcasting by the bsts-u-midas model," PhD thesis, 2015.

[10] E. Ghysels, "Mixed frequency vector autoregressive models," in *Technical Report*, mimeo, 2012.

[11] J. H. Stock and M. Watson, "Dynamic factor models," *Oxford handbook on economic forecasting*, 2011.

[12] J. Boivin and S. Ng, "Are more data always better for factor analysis?" *Journal of Econometrics*, vol. 132, no. 1, pp. 169–194, 2006.

[13] C. De Mol, D. Giannone, and L. Reichlin, "Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components?" *Journal of Econometrics*, vol. 146, no. 2, pp. 318–328, 2008.

[14] J. Bai and S. Ng, "Boosting diffusion indices," *Journal of Applied Econometrics*, vol. 24, no. 4, pp. 607–629, 2009.

[15] T. Buchen and K. Wohlrabe, "Forecasting with many predictors: Is boosting a viable alternative?" *Economics Letters*, vol. 113, no. 1, pp. 16–18, 2011.

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

[17] J. H. Stock and M. Watson, "Forecasting in dynamic factor models subject to structural instability," *The Methodology and Practice of Econometrics. A Festschrift in Honour of David F. Hendry*, vol. 173, p. 205, 2009.

[18] E. Andreou, E. Ghysels, and A. Kourtellos, "Should macroeconomic forecasters use daily financial data and how?" *Journal of Business & Economic Statistics*, vol. 31, no. 2, pp. 240–251, 2013.

[19] X. Chen and W. D. Nordhaus, "Using luminosity data as a proxy for economic statistics," *Proceedings of the National Academy of Sciences*, vol. 108, no. 21, pp. 8589–8594, 2011.

[20] N. Zhao, Y. Liu, G. Cao, E. L. Samson, and J. Zhang, "Forecasting china's gdp at the pixel level using nighttime lights time series and population images," *GIScience & Remote Sensing*, vol. 54, no. 3, pp. 407–425, 2017.

[21] C. Antoniou, G. Yannis, E. Papadimitriou, and S. Lassarre, "Relating traffic fatalities to gdp in europe on the long term," *Accident Analysis & Prevention*, vol. 92, pp. 89–96, 2016.

[22] M. Ellington, C. Florackis, and C. Milas, "Liquidity shocks and real gdp growth: Evidence from a bayesian time-varying parameter var," *Journal of International Money and Finance*, vol. 72, pp. 93–117, 2017.

[23] 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.

[24] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural networks*, vol. 1, no. 4, pp. 339–356, 1988.

[25] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modelling," Tech. Rep., 1987.

[26] G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*, Springer, 2012, pp. 62–77.

[27] S. Eickmeier and T. Ng, "Forecasting national activity using lots of international predictors: An application to new zealand," *International Journal of Forecasting*, vol. 27, no. 2, pp. 496–511, 2011.

[28] G. Kapetanios and F. Papailias, "Big data & macroeconomic nowcasting: Methodological review," *ESCoE Discussion Paper*, no. 12, 2018.

[29] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah, "A review on time series forecasting techniques for building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 902–924, 2017.

[30] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[31] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[32] V. Cho, "Tourism forecasting and its relationship with leading economic indicators," *Journal of Hospitality & Tourism Research*, vol. 25, no. 4, pp. 399–420, 2001.

[33] C. Goh and R. Law, "Modeling and forecasting tourism demand for arrivals with stochastic nonstationary seasonality and intervention," *Tourism management*, vol. 23, no. 5, pp. 499–510, 2002.

[34] K. Kandananond, "Forecasting electricity demand in thailand with an artificial neural network approach," *Energies*, vol. 4, no. 8, pp. 1246–1257, 2011.

[35] S. R. Twanabasu and B. A. Bremdal, "Load forecasting in a smart grid oriented building," 2013.

[36] J. Zhuang, Y. Chen, X. Shi, and D. Wei, "Building cooling load prediction based on time series method and neural networks," *International Journal of Grid and Distributed Computing*, vol. 8, no. 4, pp. 1386–1390, 2015.

[37] R. Nau, "Notes on the random walk model," *Fuqua School of business, Duke University. 19p*, 2014.

[38] R. Adhikari and R. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1441–1449, 2014.

[39] A. Timmermann and C. W. Granger, "Efficient market hypothesis and forecasting," *International Journal of forecasting*, vol. 20, no. 1, pp. 15–27, 2004.

[40] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for time-series forecasting," *Expert Systems with applications*, vol. 37, no. 1, pp. 479–489, 2010.

[41] M. V. Sewell, "The application of intelligent systems to financial time series analysis," PhD thesis, PhD dissertation, Department of Computer Science, University College London ..., 2010.

[42] Y. Sun, "Exchange rate forecasting with an artificial neural network model: Can we beat a random walk model?" PhD thesis, Lincoln University, 2005.

[43] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.

[44] C. A. Sims, "Macroeconomics and reality," *Econometrica: journal of the Econometric Society*, pp. 1–48, 1980.

[45] S. Mittnik and P. Zadrozny, "Forecasting quarterly german gdp at monthly intervals using monthly ifo business conditions data," in *Ifo survey data in business cycle and monetary policy analysis*, Springer, 2005, pp. 19–48.

[46] P. Caraiani *et al.*, "Forecasting romanian gdp using a bvar model," *Romanian Journal of Economic Forecasting*, vol. 13, no. 4, pp. 76–87, 2010.

[47] F. Schorfheide and D. Song, "Real-time forecasting with a mixed-frequency var," *Journal of Business & Economic Statistics*, vol. 33, no. 3, pp. 366–380, 2015.

[48] H. Lütkepohl, "Non-causality due to omitted variables," *Journal of Econometrics*, vol. 19, no. 2-3, pp. 367–378, 1982.

[49] A. Baffigi, R. Golinelli, and G. Parigi, "Bridge models to forecast the euro area gdp," *International Journal of forecasting*, vol. 20, no. 3, pp. 447–460, 2004.

[50] M. Schneider, M. Spitzer, *et al.*, "Forecasting austrian gdp using the generalized dynamic factor model," Tech. Rep., 2004.

[51] M. J. Artis, A. Banerjee, and M. Marcellino, "Factor forecasts for the uk," *Journal of forecasting*, vol. 24, no. 4, pp. 279–298, 2005.

[52] T. D. Matheson, "Factor model forecasts for new zealand," 2006.

[53] E. Angelini, M. Banbura, and G. Rünstler, "Estimating and forecasting the euro area monthly national accounts from a dynamic factor model," 2008.

[54] R. Ouysse, "Forecasting using a large number of predictors: Bayesian model averaging versus principal components regression," *UNSW Australian School of Business Research Paper*, no. 2013-04, 2013.

[55] G. Parigi and G. Schlitzer, "Quarterly forecasts of the italian business cycle by means of monthly economic indicators," *Journal of Forecasting*, vol. 14, no. 2, pp. 117–141, 1995.

[56] G. Bruno, T. Di Fonzo, R. Golinelli, and G. Parigi, "Short-run gdp forecasting in g7 countries: Temporal disaggregation techniques and bridge models," OECD-Eurostat Workshop on Frontiers in Benchmarking Techniques and Their ..., 2005.

[57] C. Foroni and M. Marcellino, "A comparison of mixed frequency approaches for nowcasting euro area macroeconomic aggregates," *International Journal of Forecasting*, vol. 30, no. 3, pp. 554–568, 2014.

[58] P. C. Higgins, "Gdpnow: A model for gdp'nowcasting'," 2014.

[59] C. Foroni and M. G. Marcellino, "A survey of econometric methods for mixed-frequency data," *Available at SSRN 2268912*, 2013.

[60] E. Ghysels, P. Santa-Clara, and R. Valkanov, "The midas touch: Mixed data sampling regression models," 2004.

[61] C. Schumacher, "A comparison of midas and bridge equations," *International Journal of Forecasting*, vol. 32, no. 2, pp. 257–270, 2016.

[62] C. Duarte *et al.*, "Autoregressive augmentation of midas regressions," Tech. Rep., 2014.

[63] E. Ghysels, A. Sinko, and R. Valkanov, "Midas regressions: Further results and new directions," *Econometric Reviews*, vol. 26, no. 1, pp. 53–90, 2007.

[64] M. P. Clements and A. B. Galvão, "Macroeconomic forecasting with mixed-frequency data: Forecasting output growth in the united states," *Journal of Business & Economic Statistics*, vol. 26, no. 4, pp. 546–554, 2008.

[65] R. Schier, R. Jurowetzki, and H. Raza, "Intelligent Machines in Economics," pp. 1–38,

[66] 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.

[67] S. G. Elmer, "Modern statistical methods applied to economic time series," PhD thesis, ETH Zurich, 2011.

[68] P.-H. Chen, C.-J. Lin, and B. Schölkopf, "A tutorial on ν-support vector machines," *Applied Stochastic Models in Business and Industry*, vol. 21, no. 2, pp. 111–136, 2005.

[69] S. Mullainathan and J. Spiess, "Machine learning: An applied econometric approach," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 87–106, 2017.

[70] A. Tiffin, "Seeing in the dark: A machine-learning approach to nowcasting in lebanon," 2016.

[71] V. Rajkumar, "Predicting surprises to gdp: A comparison of econometric and machine learning techniques," PhD thesis, Massachusetts Institute of Technology, 2017.

[72] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[73] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[74] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2013.

[75] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3299–3340, 2015.

[76] R. Mcdonald, M. Mohri, N. Silberman, D. Walker, and G. S. Mann, "Efficient large-scale distributed training of conditional maximum entropy models," in *Advances in Neural Information Processing Systems*, 2009, pp. 1231–1239.

[77] K. A. Prewitt, *A distribution-free theory of nonparametric regression. laszlo gyorfi, michael kohler, adam krzyzak, and harro walk*, 2003.

[78] K. Rakesh and P. N. Suganthan, "An ensemble of kernel ridge regression for multi-class classification," *Procedia Computer Science*, vol. 108, pp. 375–383, 2017.

[79] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[80] N. Cristianini, J. Shawe-Taylor, *et al.*, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[81] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[82] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using support vector machines," in *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, IEEE, 1997, pp. 511–520.

[83] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *omega*, vol. 29, no. 4, pp. 309–317, 2001.

[84] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[85] C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, and T. Gorka, "Data-driven method based on particle swarm optimization and k-nearest neighbor regression for estimating capacity of lithium-ion battery," *Applied Energy*, vol. 129, pp. 49–55, 2014.

[86] A. Kanas, "Non-linear forecasts of stock returns," *Journal of Forecasting*, vol. 22, no. 4, pp. 299–315, 2003.

[87] S. Yakowitz, "Nearest-neighbour methods for time series analysis," *Journal of time series analysis*, vol. 8, no. 2, pp. 235–247, 1987.

[88] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979.

[89] J. Andrada-Félix, F. Fernadez-Rodriguez, M.-D. Garcia-Artiles, and S. Sosvilla-Rivero, "An empirical evaluation of non-linear trading rules," *Studies in Nonlinear Dynamics & Econometrics*, vol. 7, no. 3, 2003.

[90] N. Meade, "A comparison of the accuracy of short term foreign exchange forecasting methods," *International Journal of forecasting*, vol. 18, no. 1, pp. 67–83, 2002.

[91] L.-Y. Chang and W.-C. Chen, "Data mining of tree-based models to analyze freeway accident frequency," *Journal of safety research*, vol. 36, no. 4, pp. 365–375, 2005.

[92] M. Xu, P. Watanachaturaporn, P. K. Varshney, and M. K. Arora, "Decision tree regression for soft classification of remote sensing data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 322–336, 2005.

[93] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761–1768, 2007.

[94] S. P. Curram and J. Mingers, "Neural networks, decision tree induction and discriminant analysis: An empirical comparison," *Journal of the Operational Research Society*, vol. 45, no. 4, pp. 440–450, 1994.

[95] M. R. Segal, "Machine learning benchmarks and random forest regression," 2004.

[96] J. Mei, D. He, R. Harley, T. Habetler, and G. Qu, "A random forest method for real-time price forecasting in new york electricity market," in *2014 IEEE PES General Meeting— Conference & Exposition*, IEEE, 2014, pp. 1–5.

[97] I. Barandiaran, "The random subspace method for constructing decision forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, 1998.

[98] G. G. Creamer and Y. Freund, "Predicting performance and quantifying corporate governance risk for latin american adrs and banks," 2004.

[99] D. Van den Poel and B. Lariviere, "Customer attrition analysis for financial services using proportional hazard models," *European journal of operational research*, vol. 157, no. 1, pp. 196–217, 2004.

[100] G. Lebanon and J. D. Lafferty, "Boosting and maximum likelihood for exponential models," in *Advances in neural information processing systems*, 2002, pp. 447–454.

[101] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[102] R. Avnimelech and N. Intrator, "Boosting regression estimators," *Neural computation*, vol. 11, no. 2, pp. 499–520, 1999.

[103] W. Y. Goh, C. P. Lim, and K. K. Peh, "Predicting drug dissolution profiles with an ensemble of boosted neural networks: A time series approach," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 459–463, 2003.

[104] E. Canestrelli, P. Canestrelli, M. Corazza, M. Filippone, S. Giove, and F. Masulli, "Local learning of tide level time series using a fuzzy approach," in *2007 International Joint Conference on Neural Networks*, IEEE, 2007, pp. 1813–1818.

[105] J. Heo and J. Y. Yang, "Adaboost based bankruptcy forecasting of korean construction companies," *Applied soft computing*, vol. 24, pp. 494–499, 2014.

[106] D. L. Shrestha and D. P. Solomatine, "Experiments with adaboost. rt, an improved boosting scheme for regression," *Neural computation*, vol. 18, no. 7, pp. 1678–1710, 2006.

[107] N. Duffy and D. Helmbold, "Boosting methods for regression," *Machine Learning*, vol. 47, no. 2-3, pp. 153–200, 2002.

[108] R. Mitchell, A. Adinets, T. Rao, and E. Frank, "Xgboost: Scalable gpu accelerated learning," *arXiv preprint arXiv:1806.11248*, 2018.

[109] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[110] J. Huang and M. Perry, "A semi-empirical approach using gradient boosting and k-nearest neighbors regression for gefcom2014 probabilistic solar power forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1081–1086, 2016.

[111] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 308–324, 2015.

[112] N. E. Johnson, O. Ianiuk, D. Cazap, L. Liu, D. Starobin, G. Dobler, and M. Ghandehari, "Patterns of waste generation: A gradient boosting model for short-term waste prediction in new york city," *Waste management*, vol. 62, pp. 3–11, 2017.

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

[114] R. Mitchell and E. Frank, "Accelerating the xgboost algorithm using gpu computing," *PeerJ Computer Science*, vol. 3, e127, 2017.

[115] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of p2p network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning," *Electronic Commerce Research and Applications*, vol. 31, pp. 24–39, 2018.

[116] T. Lei, F. Chen, H. Liu, H. Sun, Y. Kang, D. Li, Y. Li, and T. Hou, "Admet evaluation in drug discovery. part 17: Development of quantitative and qualitative prediction models for chemical-induced respiratory toxicity," *Molecular pharmaceutics*, vol. 14, no. 7, pp. 2407–2421, 2017.

[117] A. Torres-Barrán, Á. Alonso, and J. R. Dorronsoro, "Regression tree ensembles for wind energy and solar radiation prediction," *Neurocomputing*, vol. 326, pp. 151–160, 2019.

[118] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[119] M. Mehta, R. Agrawal, and J. Rissanen, "Sliq: A fast scalable classifier for data mining," in *International conference on extending database technology*, Springer, 1996, pp. 18–32.

[120] J. Shafer, R. Agrawal, and M. Mehta, "Sprint: A scalable parallel classifier for data mining," in *VLDB*, Citeseer, vol. 96, 1996, pp. 544–555.

[121] S. Ranka and V. Singh, "Clouds: A decision tree classifier for large datasets," in *Proceedings of the 4th Knowledge Discovery and Data Mining Conference*, vol. 2, 1998, p. 8.

[122] R. Jin and G. Agrawal, "Communication and memory efficient parallel decision tree construction," in *Proceedings of the 2003 SIAM International Conference on Data Mining*, SIAM, 2003, pp. 119–129.

[123] P. Li, Q. Wu, and C. J. Burges, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *Advances in neural information processing systems*, 2008, pp. 897–904.

[124] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, Springer, 2000, pp. 1–15.

[125] C. Chatfield, "What is the 'best'method of forecasting?" *Journal of Applied Statistics*, vol. 15, no. 1, pp. 19–38, 1988.

[126] G. M. Jenkins, "Some practical aspects of forecasting in organizations," *Journal of forecasting*, vol. 1, no. 1, pp. 3–21, 1982.

[127] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *Journal of forecasting*, vol. 1, no. 2, pp. 111–153, 1982.

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

[129] S. Makridakis, C. Chatfield, M. Hibon, M. Lawrence, T. Mills, K. Ord, and L. F. Simmons, "The m2-competition: A real-time judgmentally based forecasting study," *International Journal of Forecasting*, vol. 9, no. 1, pp. 5–22, 1993.

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

[131] K. A. Aastveit, K. R. Gerdrup, and A. S. Jore, "Short-term forecasting of gdp and inflation in real-time: Norges bank's system for averaging models," 2011.

[132] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. B. Hamrick, J. Grout, S. Corlay, *et al.*, "Jupyter notebooks-a publishing format for reproducible computational workflows.," in *ELPUB*, 2016, pp. 87–90.

[133] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[134] V. Kuznetsov and M. Mohri, "Learning theory and algorithms for forecasting non-stationary time series," in *Advances in neural information processing systems*, 2015, pp. 541–549.