DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

META LEARNING FOR TIME SERIES FORECASTING

by

Uğur Baran YAPAR

October, 2023

İZMİR

META LEARNING FOR TIME SERIES FORECASTING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for Master of Science in
Department of Statistics, Statistics Program

by

Uğur Baran YAPAR

October, 2023

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"META LEARNING FOR TIME SERIES FORECASTING"** completed by **UĞUR BARAN YAPAR** under supervision of **ASSOC.PROF.DR. İDİL YAVUZ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc.Prof.Dr. İdil YAVUZ

Supervisor

Doç. Dr. Özgül VUPA ÇİLİNGİROĞLU

Jury Member

Dr. Öğr. Üyesi Ümit KUVETLİ

Jury Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who have supported and contributed to the completion of this thesis. Without their guidance, encouragement, and assistance, this work would not have been possible.

First and foremost, I am deeply thankful to my thesis supervisor, Assoc. Prof. Dr. İdil YAVUZ, for their invaluable mentorship and unwavering support. Their expertise and insightful feedback have been instrumental in shaping this research.

I am grateful to TUBİTAK for providing financial support that allowed me to focus on my research without financial constraints.

I cannot overlook the support of my family, who have been my pillars of strength and unwavering encouragement. Their belief in my abilities kept me motivated during the ups and downs of this academic endeavor.

I am indebted to the participants who generously shared their time and insights for the research. Their cooperation was invaluable in gathering the data necessary for this thesis.

Once again, thank you to everyone who played a part, big or small, in making this thesis a reality. Your support has been immensely valued and appreciated.

Uğur Baran YAPAR

META LEARNING FOR TIME SERIES FORECASTING

ABSTRACT

Accurate forecasting is very important for both individuals and institutions and in every field of science, so there are many studies in the literature. However, at the moment, two major methods stand out, Exponential Smoothing (ES) and Box-Jenkins ARIMA. Comparing these two major method, ES is quite simple but ARIMA is complex and difficult. However, the main reason that makes the ES method popular in use is its proven success in forecasting performance. Despite the popularity of the ES method, there are still some unsolved problems such as initial value problem. Until recently, an alternative method could not be developed to these two basic methods. Ata method is an innovative new forecasting technique where the forms of the models are similar to exponential smoothing models but the smoothing parameters depend on the sample size, are optimized on a discrete space and initialization is both easier as it is done simultaneously when the parameters are optimized and is less influential since the weights assigned to initial values approach zero quickly. Ata can be applied to all time series settings easily and provides better forecasting performance due to its flexibility. The success of the Ata forecasting method in the empirical and simulated data sets is no coincidence. Recently, different methods and algorithms have been used in the the literature to improve forecasting performance. Prominent among these methods are machine learning, meta learning and different combination methods. In this study, the current forecasting performance of the Ata method will be improved by using different trials and meta learning technique using M3 competition data sets.

Keywords: Ata method, Exponential smoothing, Time series, Forecasting, Meta learning

ZAMAN SERİLERİNDE ÖNGÖRÜMLEME İÇİN META ÖĞRENME

ÖZ

Doğru tahmin hem bireyler hem de kurumlar açısından ve bilimin her alanında oldukça önemli olduğundan literatürde pek çok çalışma bulunmaktadır. Ancak şu anda iki ana yöntem öne çıkıyor; Üstel Düzeltme (ES) ve Box-Jenkins ARIMA. Bu iki ana yöntemi karşılaştırdığımızda ES oldukça basittir ancak ARIMA karmaşık ve zordur. Ancak ES yöntemini popüler hale getiren temel neden performans tahmininde kanıtlanmış başarısıdır. ES yönteminin popülaritesine rağmen başlangıç değer problemi gibi hala çözülmemiş problemler bulunmaktadır. Yakın zamana kadar bu iki temel yönteme alternatif bir yöntem geliştirilememişti. Ata yöntemi, modellerin formlarının üstel yumuşatma modellerine benzer olduğu ancak yumuşatma parametrelerinin örneklem boyutuna bağlı olduğu, ayrı bir alanda optimize edildiği ve başlatmanın, parametreler eş zamanlı olarak yapıldığından daha kolay olduğu yenilikçi yeni bir tahmin tekniğidir. optimize edilmiştir ve başlangıç değerlerine atanan ağırlıklar hızla sıfıra yaklaştığından daha az etkilidir. Ata, tüm zaman serisi ayarlarına kolaylıkla uygulanabilmekte ve esnekliği sayesinde daha iyi tahmin performansı sağlamaktadır. Ata tahmin yönteminin ampirik ve simüle edilmiş veri setlerindeki başarısı tesadüf değildir. Son zamanlarda literatürde tahmin performansını artırmak için farklı yöntem ve algoritmalar kullanılmaktadır. Bu yöntemler arasında öne çıkanlar makine öğrenmesi, meta öğrenme ve farklı kombinasyon yöntemleridir. Bu çalışmada M3 yarışma veri setleri kullanılarak farklı denemeler ve meta öğrenme tekniği kullanılarak Ata yönteminin mevcut tahmin performansı iyileştirilecektir.

Anahtar kelimeler: Ata metod, Üssel düzleştirme, Zaman serisi, Öngörümleme, Meta öğrenme

CONTENTS

Page

THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGMENT iii
ABSTRACT iv
ÖZ v
CONTENTS vi
LIST OF TABLES vii

CHAPTER ONE - INTRODUCTION 1

CHAPTER TWO - OVERVIEW OF TIME SERIES 7

2.1 Existing Methods ……..

2.2 Error Measures ……..

CHAPTER THREE – ATA METHOD 16

CHAPTER FOUR – META LEARNING 20

4.1 Meta Learning for Forecasting……...21

4.2 Simple Averaging……...22

4.3 Using the Median of Forecasts……...24

4.4 Time Series Features……..28

4.5 Feature Based Meta Learning Approaches for Time Series Forecasting…...29

4.6 Binary Logistic Regression……29

4.7 Classification Trees……32

CHAPTER FIVE – APPLICATION……...35

CHAPTER SIX – RESULTS AND CONCLUSIONS……41

LIST OF TABLES

Table 2.1 Categorization of ES methods by trend and seasonality…….....11

Table 4.1 Specific Features.....26

Table 4.1 Continued Specific Features.....27

Table 4.1 Continued Specific Features.....28

Table 4.2 An Example Confusion Matrix.....31

Table 4.3 Performane Measures.....32

Table 5.1 Average symmetric MAPE of ATA(p,0) for different accuracy measures...36

Table 5.2 Average symmetric MAPE of Ata(p, 1) for different accuracy measures....37

Table 5.3 Average symmetric MAPE of combination Ata(p, 0) and Ata(p, 1) for different accuracy measures.....37

Table 5.4 Accuracies of learners on identifying optimal forecaster.....39

CHAPTER ONE
INTRODUCTION

Time series data is a collection of data that occurs when observations are recorded chronologically within equal time intervals. These time intervals can be annual, weekly, daily, hourly and even seconds for the purpose at hand. Although technical analysis of time series has become widespread with the development of today's technology, its beginnings go back a long way. In the past, humanity used time series to record the movements of planets, but today it is often used in almost every subject. Annual changes in a country's population, weekly sales figures of a company, the amount of daily rainfall falling on a city, sudden changes in a patient's blood values and so on can be statistically analyzed and predicted for the future. Time series forecasting is the prediction of future movements under the assumption that future movements will be similar to past tendencies using historical data. Trend, seasonality, cyclicality and irregularity, which are components of the time series, can be cited as examples of these repetitive movements. Making accurate forecasts is of great importance in all fields of science. Especially, modelling big data stored easily with the development of computers and turning them into accurate forecast has become vital in fields like industry, finance and medicine. Accurate forecast prevent waste of resources and loss of opportunity and provide economic gains.

The fundamental aim of time series is to observe the variable of interest over time and predict the future using the observed values. Although there are numerous studies in the literature on how to do this, the basic assumption is that the future will be a repetition of the past. Despite the numerous studies on predicting the future, there are still two basic methods as of now. These are Box-Jenkins ARIMA and Exponential Smoothing (De Gooijer & Hyndman, 2005). Numerous studies have been conducted on Box-Jenkins ARIMA and ES methods in the forecasting literature, but ES method is more popular than Box-Jenkins ARIMA method (Goodwin, 2010). This is because the ES method is simpler than the Box-Jenkins ARIMA methods, but more importantly, it has been more successful in forecasting performance on competitive data sets in the past (Makridakis et al., 1984; Makridakis & Hibon, 2000; Koning et al., 2005) .

The search for new alternative methods to these two basic methods has been continuous in the literature, but it has been found that all proposed methods are special cases of these two basic methods and their performance is also specific to certain data sets. Only in the M3 forecasting competition held in 2000 (Makridakis & Hibon, 2000), a special method that is not included in the literature drew attention with its first place performance: the THETA method (Assimakopoulos & Nikolopoulos, 2000). However, It was (Hyndman & Billah, 2003) proved that this method was not a new method, but a special case of the ES method, but it is now accepted as a statistical benchmark method in competitions due to its success. Theta (Hyndman & Billah, 2003) is a special case of simple exponential smoothing with drift. This model involves fitting two -lines, forecasting the lines using a Simple Exponential Smoother, and then combining the forecasts from the two lines called “Theta lines” to produce the final forecast. With its simple structure and predictive power, it performs well when forecasting in time series of various structures. Theta led the M3 competition with the 3003 time series, but did not perform the same in the M4 (Makridakis et al., 2018) competition.

Although ES methods have been theoretically substantiated by the state-space framework developed by Hyndman et al. (2002) and the calculation of forecasting intervals has been enabled, they still complain about problems that negatively affect forecasting performance. The first of these is referred to as the "initial value problem" in the literature. For example, in simple ES, the initial value problem arises especially for large values in relatively short time series (Gardner, 2006). In this case, since too much weight is given to the initial value, which is also a predicted value, the selection of the initial value becomes very critical. This situation makes the forecasting process inconsistent and very sensitive to initial value selection, which is an unwanted situation. Another problem is the sensitivity of ES methods to outliers. Although there are methods to deal with outliers in the literature, these methods can also negatively affect forecasting performance.

ES is undisputedly one of the most widely used forecasting methods available due to its simplicity, adaptiveness and accuracy (Goodwin, 2010). ES puts less emphasis on old data than new data, assuming that old data should have less impact on the

forecast than current data. In other words, the weight of observations increases as we move forward in time. The main purpose of exponential smoothing is to use flattened data to predict future values by smoothing the original series. There are three main types of exponential smoothing: Simple Exponential Smoothing (Brown, 1959), Holt Exponential Smoothing (Holt, 1957) and Holt-Winters Exponential Smoothing (Winters, 1960). Simple Exponential Smoothing is a commonly used forecasting method that does not contain trend or seasonality components, but only includes a smoothing parameter alpha. In addition, Holt Exponential Smoothing includes the trend component, while Holt-Winters Exponential Smoothing can cover trend and seasonal components.

In addition to the well-established models that have been included in the time series prediction, Ata (Yapar et al., 2017), an alternative forecasting method that has just entered the literature, will be included in this thesis. Ata, which can overcome a number of disadvantages of ES, especially the initialization problem, gives successful results in predicting complex data sets with its simple structure, speed and performance.

There can be no guarantee that any forecasting method in the literature will produce accurate forecasts for all data sets. No matter how complex and advanced the methods are, they cannot produce accurate forecasts for some data types. Two basic approaches can be considered to forecast time series (Talagala et al., 2018). We can either apply the same forecast method to all time series, or we can apply individual forecast methods for each series. However, when we deal with a large number of data, it is inevitable that these primitive methods will put us at a disadvantage in terms of both accuracy and time. Instead of using these methods, we aim to produce an algorithm in which we can get fast and accurate results by considering the characteristics of each time series and selecting methods according to these characteristics. We will call these characteristics "features". A feature is a measurable property or parameter of the data-set. We can give the features an example: the strength of trend, the strength of seasonality, spikiness, linearity, spectral entropy, autocorrelation value etc. Fulcher & Jones (2014) identified 9.000 operations to extract features from time series.

Machine learning is an artificial intelligence application that does not need explicit programming to fulfill the desired instructions, but instead improves itself using past experiences. The fact that machine learning can improve by learning from historical data and experience is similar to the way people learn. Machine learning saves time and achieves reliable results by creating models that can automatically quickly analyze large amounts of complex data. In general, machine learning algorithms are used for a prediction or classification problem. The model to be built must have plenty of data and go through a large number of training stages. These algorithms try to predict new output values by using historical data as inputs. It tries to reach the best model with the lowest error value in candidate models and supports it with the help of optimization. Today, many large companies such as Facebook, Google and Amazon use these methods to process millions of data quickly.

Machine learning algorithms are mainly divided into supervised machine learning, unsupervised machine learning and reinforcement algorithms. Supervised machine learning needs a developer's control to predict future events. The developer tries to reach the output variable in the most accurate way by tagging the data used in the training and setting certain rules for the algorithms to be installed. By processing the available data repeatedly, the system learns the relationship between the data and reaches a conclusion. Estimating house prices or determining whether the e-mails received in our e-mail are spam is obtained by supervised algorithm methods. Unsupervised machine learning is used in cases where data is not labeled during the training phase. The main purpose of this method, which does not need the intervention of the developer and learns the natural structure from the input data, is to separate the data at certain limits by associating it with each other. Unsupervised machine learning is used in customer segmentation and image processing systems. K-means clustering method and principal component analysis are the methods used in this method. Reinforcement algorithms make inferences by learning from the mistakes they make as they move to the intended result and reach the result with minimal error. It adopts a structure similar to the reward-penalty system. It records the events that occur by interacting with its surroundings while achieving the correct result and maximizes its performance by receiving feedback from the mistakes made. An example of this system is that a person gets negative feedback from his surroundings because of

burning his mouth due to eating hot peppers and does not repeat the same mistake again. The most commonly used supervised machine learning methods are used to classify or predict data using tagged datasets. This method includes an array of input variables and an output variable. The algorithm to be built ensures that input and output variables are mapped and trained on the data set and modified until it reaches an acceptable level of performance. Some methods used in supervised learning include linear regression, logistical regression, random forest, support vector machine (SVM), neural networks and more. Supervised learning; regression problems and classification problems are divided into two categories. In the regression problem, the model is used to predict future values by training with historical data. In the classification problem, various tags train the algorithm to identify items in a specific category. While estimating the price of a product in the future is an example of regression problem, determining whether a person in the hospital is sick or healthy is an example of a classification problem.

Meta-learning, which emerged in the 1980s through the work of Jürgen Schmidhuber and Yoshua Bengio, is defined as learning to learn and is a subset of machine learning. Meta-learning aims to perform a faster learning process when encountering unknown data sets by observing the performance of different machine learning models. For example, if a person is repeatedly trained to ride a bike, he or she can find a way to ride a motorcycle on his own using his past experience. Meta-learning algorithms can similarly learn to use the best forecasts from machine learning algorithms to make better forecasts. Meta-learning techniques help train and optimize machine learning models to address existing challenges. Model selection in time series using machine learning techniques is a process that continues to develop from the past to the present. In this process, the relevant study is treated as a classification problem and machine learning algorithms are established using features used as input values.

In order to produce accurate and good predictions when studying machine learning, it is necessary to have plenty of data when training our model. For this reason, we carry out our work with 3.003 time series used in the M3-Competition, the most up-to-date of the famous M-Competiton based on time series forecasting. The actual 3003 time series we have available consists of 6 different areas (micro, industy, macro, finance,

demographic, other) and 6 different time periods (yearly, quarterly, monthly, weekly, daily, hourly). The purpose of the competition is to generate predictions using previously described time series (input data). As a result of comparing the actual values (output values) described later by the organizers with the forecast values calculated by the contestants, forecast errors are calculated and models with the lowest value are determined as the best. The winner of the M3 competition was Theta (Assimakopoulos & Nikolopoulos, 2000), but failed to perform in the M4 competition. This is due to the fact that not every method is always able to adapt to the series collection. Otherwise, we would be able to easily estimate every data with the best single method in the literature. The performance of ATA method, which has been proven to be better adapted to different types of time series data, will be taken into consideration in this study.

In our study, we considered the estimate of 3.003 time series data in the M3 competition as a supervised machine learning problem and adapted meta-learning techniques through "features" to the problem of selecting time series models. In line with the results obtained with meta-learning, it is aimed to give suggestions about model and method selection and to contribute to the forecasting literature. All of the time series forecasting methods can be considered, but especially exponential smoothing, Box-Jenkins, Theta, ATA and artificial neural networks will be used. In the meta learning step, decision trees, artificial neural networks and decision support machines and the recently popular XGBoosting algorithm will be used.

CHAPTER TWO
OVERVIEW OF TIME SERIES

2.1 Existing Methods

Time series data emerges in very different fields, when a random variable is observed over time. The main purpose in time series is to observe the variable of interest over time and to predict the future using the observed values. If the observed values are represented by y_t, $t = 1,2,3,\ldots n$. The aim is to produce predictive values for future observations. There are numerous studies in the literature on how this can be done, but the basic assumption is that the future will be a repetition of the past.

In the literature, firstly, studies have been carried out for time series oscillating with a random noise around only one level where e_t represent a random noise with zero mean and constant variance:

$$y_t = a + e_t \tag{2.1}$$

In time series modeled in this way, only estimating level *a* is sufficient to predict the future. That is, if *h* is represented by step forward prediction. The general form of the estimates to be produced for the level is as follows (Brown, 1959):

$$\hat{a} = F(y_1,\ldots,y_n) = \sum_{t=1}^{n} w_t y_t \tag{2.2}$$

Here, w_t, $w_t \in [0,1]$, $\sum_{i=1}^{n} w_t = 1$ represents the weights. Different weight vectors can be selected to shape the estimation method. When choosing these methods, methods that give more weight to current observations are preferred with the thought that the recent past will predict the future better, and whether a method weights current observations effectively or not is possible by calculating the average age (*AA*) for that weighting. The average age is calculated as follows:

$$AA = n - \sum_{t=1}^{n} t w_t \tag{2.3}$$

In addition to giving more weight to current observations, a good method would also be expected to be able to estimate the level with a small variance. The prediction variance is found as follows, assuming the constant variance of the random noise is σ^2

$$Var\left(\hat{a}\right) = \sum_{t=1}^{n} w_t^2 \sigma^2 = \sum_{t=1}^{n} w_t^2 \sigma^2 = V\sigma^2 \tag{2.4}$$

Although small values of these two measurements are preferred, since one decreases mathematically and the other increases, establishing a balance between them plays an important role in the selection of weighting.

Although they are extreme examples in practice, two approaches with successful results are still used. The first of these is known as the mean method, and in this method, all observations are weighted equally when estimating the level, that is $\hat{a} = \bar{y}$ and for this method

$$AA = \frac{(n-1)}{2}, \quad V = \frac{1}{n} . \tag{2.5}$$

Another extreme method is called the naive method, and when this method is used, all weight is given to the last observation when estimating the level, $\hat{a} = y_n$ and for this method.

$$AA = 0, \quad V = 1. \tag{2.6}$$

Another popular method is the moving average method, which is located between these two extreme methods. This method gives equal weight to *p* current observations, which is the parameter value, and gives zero weight to all of the older observations. So the level

$$\hat{a} = \frac{y_n + y_{n-1} + . \quad . \quad . + y_{n-p+1}}{p} = \frac{1}{p}\sum_{j=0}^{p-1} y_{n-j} \quad , \tag{2.7}$$

and

$$AA = \frac{(p-1)}{2}, \quad V = \frac{1}{p}. \tag{2.8}$$

The simple exponential smoothing (SES) method, which is based on the idea that current observations are more important when predicting the future by making use of all observations, and in which the observations are weighted with exponentially decreasing weights from the most recent observation to the oldest observation, also has a very important place in the literature. In this method,

$$l_t = \alpha y_t + (1-\alpha) l_{t-1}. \tag{2.9}$$

l_t, represent the smoothed value at time *t*, namely the level, and $\alpha \in [0,1]$ smoothing parameter. When the smoothing process is repeated retrospectively, it is seen that

$$l_t = \alpha \sum_{k=0}^{t-1} (1-\alpha)^k y_{t-k} + (1-\alpha)^t l_0 \ . \tag{2.10}$$

As can be seen here, in addition to a smoothing parameter, the model also needs an initial value, l_0. Here also, it is clearly seen that the future is a weighted average of the past, current observations are given more weight than previous observations, and these weights decrease exponentially as one goes to the past. For this method,

$$AA = \frac{(1-\alpha)}{\alpha}, \quad V = \frac{\alpha}{2-\alpha}. \tag{2.11}$$

Holt (1957) extended the idea of simple exponential smoothing in his work, allowing trended data to be estimated as well. This method uses two smoothing (for level and trend) and a prediction equation:

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} - b_{t-1}), \tag{2.12}$$

$$b_t = \beta (l_t - l_{t-1}) + (1-\beta) b_{t-1}, \tag{2.13}$$

$$\hat{y}_t(h) = l_t + h b_t. \tag{2.14}$$

In the above method, in addition to simple exponential smoothing, the smoothing parameter $\beta \in [0,1]$ and its initial value b_0 determined. In total, two initial values must be determined and two parameters optimized.

In a different version of the exponential smoothing method, which includes the additive trend, which is also called Holt's linear trend method in the literature, the level and trend can be modeled multiplicatively (Gardner, 1985). This method can be described by the following equations:

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} * b_{t-1}), \tag{2.15}$$

$$b_t = \beta(l_t / l_{t-1}) + (1-\beta)b_{t-1}, \tag{2.16}$$

$$\hat{y}_t(h) = l_t * b_t^h. \tag{2.17}$$

In exponential smoothing methods that include multiplicative and additive trends, while producing forward forecasts, as h gets larger, in some cases, the effect of the trend grows too large and estimations begin to be obtained far beyond the course of the data. To cope with this situation, exponential smoothing method with damped trend is proposed for both additive and multiplicative methods.

The additive trend model takes the following form when the damped trend effect is added (Gardner & McKenzie (1985)):

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} - \phi b_{t-1}), \tag{2.18}$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)\phi b_{t-1}, \tag{2.19}$$

$$\hat{y}_t(h) = l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t. \tag{2.20}$$

Here, ϕ is called the damping parameter and when it is equal to 1, the damping effect is eliminated.

The model with a multiplicative trend takes the following form when the damped trend effect is added (Taylor (2003)):

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} * b_{t-1}^{\phi}), \tag{2.21}$$

$$b_t = \beta(l_t / l_{t-1}) + (1-\beta)b_{t-1}^{\phi}, \tag{2.22}$$

$$\hat{y}_t(h) = l_t * b_t^{(\phi+\phi^2+\cdots+\phi^h)} . \quad (2.23)$$

Again in this model, when the damping parameter is equal to 1, the model is reduced to an exponential smoothing model with a multiplicative trend.

The categorization of exponential smoothing methods was first proposed by Pegels in 1969. This decomposition was later extended by Gardner and McKenzie (1985) and the additive damped trend method was added to them, and it was completed by adding the multiplicative damped trend method with the study of Taylor (2003). Five types of trends (absent, additive, multiplicative, additive damped, multiplicative damped) and three types of seasonality (absent, additive, multiplicative) were considered in the decomposition of these categories, and the following 15 models were examined.

Table 2.1 Categorization of ES methods by trend and seasonality.

	Seasonality		
Trend	None (Y)	Additive(A)	Multiplicative (M)
None (N)	(N,N)	(N,A)	(N,M)
Additive (A)	(A,N)	(A,A)	(A,M)
Additive Dumped (Ad)	(Ad,N)	(Ad,A)	(Ad,M)
Multiplicative (M)	(M,N)	(M,A)	(M,M)
Multiplicative Dumped(Md)	(Md,N)	(Md,A)	(Md,M)

Formulas and estimation equations for all these 15 models are given by Hyndman and Athanasopoulos (2014), and suggestions for initial value selection are also made in the same study.

Hyndman et al. (2002) created an innovative state space framework that encompasses all exponential smoothing methods and enables the creation of predictive intervals and likelihood functions for all these methods, thus facilitating model selection. By expanding this framework (Hyndman et al. (2008)), each model in Table 1 was written in two ways, thus increasing the number of possible models to 30, allowing the error component to be added to the model as well. For these models, the notation ETS(*,*,*) is used, and the triplet (*,*,*) in this notation represents the possible modeling of the error (E), trend (T) and seasonality (S) components, respectively. The representations of these possible models in the form of state space are clearly given in the study of Hyndman et al. (2002). A program that enables the optimization of these models and the selection of models among them was developed in the R programming language and presented to users in the "forecast" package (Hyndman et al., (2008)).

Although the exponential smoothing methods are theoretically strengthened by the state-space framework developed by Hyndman et al. still complains problems that negatively affect forecasting performance. The first of these is called the initial value problem in the literature. For example, in simple exponential smoothing we encounter the initial value problem, especially for relatively short time series for small α values. In this case, the first value selection becomes very critical since too much weight will be given to the first value, which is itself a predictive value. This is undesirable as it makes the estimation process inconsistent and very sensitive to the initial value selection. Another problem arises in parameter optimization when using exponential smoothing methods. The higher the degree of exponential smoothing model, the higher the number of initial values and parameters that need to be optimized. This optimization problem becomes complex and time consuming since parameter estimations are made in continuous space and many parameters and initial values are needed. In addition, according to the error type used, different parameter values and

different models appear as the best model for the user, which makes the usability of the method difficult.

Similar problems apply to the Box-Jenkins ARIMA method. First of all, although they have a very solid statistical infrastructure, the establishment of Box-Jenkins ARIMA models requires expertise, and the model to be recommended as the best model may vary from expert to expert. For this reason, the use of ARIMA in estimation cannot be automated and this method, which requires time and expertise, cannot cope with the large data flow that requires fast and healthy analysis and interpretation.

While modeling the time series, it is thought that the most appropriate model is the one that best fits the in-sample data obtained, and it is assumed that the predictions to be obtained from this model will be the closest estimates with the idea that the future will be a repetition of the past. After a model is established, the performance of that model is evaluated according to how well it predicts the new values (out-sample) of the observed time series (Makridakis and Hyndman (1998)).

During the model building phase, many different error types have been proposed to measure the fit with the available data set and to evaluate the performance of the model. After $e_t = y_t - y_t$ being defined as an error for any moment, these error types can be grouped under four main headings.

2.2 Error Measures

Error measures are an important component of time series analysis. A key aspect of time series analysis is the evaluation of the accuracy of forecasts made using models fitted to the data. In order to evaluate the accuracy of a forecast, error measures are used. Error measures provide a quantitative measure of the deviation between actual and predicted values $e_t = \left(y_t - \hat{y}_t\right)$. The choice of error measure depends on the specific application and the properties of the data. The mean squared error, mean absolute error, root mean squared error, mean absolute percentage error, and symmetric mean absolute percentage error are among the most commonly used error measures in time series analysis. Each of these error measures has its own advantages

and disadvantages, and the choice of error measure should be based on a careful evaluation of the data and the specific application (Hyndman et al., 2006).

Mean Squared Error (MSE): The mean squared error (MSE) is a widely used error measure in time series analysis. The MSE is defined as the average of the squared differences between the actual and predicted values.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2 \ . \tag{2.24}$$

It provides a measure of the overall accuracy of a forecast. The MSE is a popular choice because it is easy to compute and has many desirable properties. One of the main advantages of the MSE is that it penalises large errors more heavily than small errors.

Mean Absolute Error (MAE): The mean absolute error (MAE) is another commonly used error measure in time series analysis. The MAE is defined as the average of the absolute differences between the actual and predicted values.

$$MAE = \frac{1}{n}\sum_{t=1}^{n} \left| e_t \right| \ . \tag{2.25}$$

The MAE is a robust error measure because it is not affected by outliers. However, the MAE does not penalise large errors more heavily than small errors, which can be a disadvantage in some applications.

Root Mean Squared Error (rMSE): The root mean squared error (rMSE) is a popular alternative to the MSE. The rMSE is defined as the square root of the MSE.

$$rMSE = \sqrt{MSE} \ . \tag{2.26}$$

The advantage of the rMSE over the MSE is that it is in the same units as the data, which makes it easier to interpret. The rMSE is also more sensitive to large errors than the MSE, which can be an advantage in some applications. These measures are scale-dependent measures and cannot be used to make comparisons between on different scales. They can be used to compare different methods applied the same data set.

Percentage errors have the advantage of being scale independent and they can be used to compare performance between different data series.

Mean Absolute Percentage Error (MAPE): The mean absolute percentage error (MAPE) is a popular error measure for forecasting applications. The MAPE is defined as the average of the absolute percentage differences between the actual and predicted values.

$$MAPE = 100\frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{|y_t|}. \tag{2.27}$$

The MAPE is expressed as a percentage, which makes it easy to interpret. The MAPE has the disadvantage that it is not well-defined when the actual value is zero.

Symmetric Mean Absolute Percentage Error (sMAPE): The symmetric mean absolute percentage error (sMAPE) is an alternative to the MAPE that addresses the issue of zero actual values. The sMAPE is defined as the average of the absolute percentage differences between the actual and predicted values, normalised by the sum of the actual and predicted values.

$$sMAPE = 100\frac{1}{n}\sum_{t=1}^{n}\frac{\left|e_t \middle/ 2\right|}{|y_t + \hat{y}_t|}. \tag{2.28}$$

The sMAPE is symmetric because it treats over-predictions and under-predictions equally. The sMAPE is also expressed as a percentage, which makes it easy to interpret.

Scaled errors were proposed by Hyndman & Kohler (2006) as an alternative to percentage errors when comparing forecast accuracy on different scales. For a non-seasonal time series, a useful way to use mean absolute scale error (MASE):

$$q_t = \frac{e_t}{\frac{1}{n}\sum_{t=1}^{n}|y_t - y_{t-1}|}, \tag{2.29}$$

$$MASE = mean|q_t|. \tag{2.30}$$

CHAPTER THREE
ATA METHOD

Ata method is an innovative new forecasting technique where the forms of the models are similar to exponential smoothing models but the smoothing parameters depend on the sample size, are optimized on a discrete space and initialization is both easier as it is done simultaneously when the parameters are optimized and is less influential since the weights assigned to initial values approach zero quickly (Yapar et al., 2019; Taylan et al., 2021). ATA can be applied to all time series settings easily and provides better forecasting performance due to its flexibility (Yapar et al., 2017).

For a time series $\{y_1, \dots y_n\}$ ATA method can be given in additive form as below:

$$l_t = \left(\frac{p}{t}\right) y_t + \left(\frac{t-p}{t}\right)(l_{t-1} + \emptyset b_{t-1}), \tag{3.1}$$

$$\mathrm{b_t} = \left(\frac{\mathrm{q}}{\mathrm{t}}\right)(\mathrm{l_t} - \mathrm{l_{t-1}}) + \left(\frac{\mathrm{t-q}}{\mathrm{t}}\right)(\emptyset \mathrm{b_{t-1}}), \tag{3.2}$$

where p is the smoothing parameter for level, q is the smoothing parameter for trend, $\emptyset$ is the dampening parameter and $l_t = y_t$ for $t \leq p$, $b_t = y_t - y_{t-1}$ for $t \leq q$, $b_1 = 0$, $p \in \{1,2, \dots, n\}$, $q \in \{0,1,2, \dots, p\}$, $\emptyset \in (0,1]$. Then, the h step ahead forecasts can be obtained by:

$$\hat{y}_{t+h|t} = l_t + (\emptyset + \emptyset^2 + \cdots + \emptyset^h) b_t. \tag{3.3}$$

Similarly for a time series $\{y_1, \dots y_n\}$ ATA method can be given in multiplicative form as below:

$$l_t = \left(\frac{p}{t}\right) y_t + \left(\frac{t-p}{t}\right)\left(l_{t-1} b_{t-1}^{\emptyset}\right), \tag{3.4}$$

$$b_t = \left(\frac{q}{t}\right)\left(\frac{l_t}{l_{t-1}}\right) + \left(\frac{t-q}{t}\right)\left(b_{t-1}^{\emptyset}\right), \tag{3.5}$$

where again p is the smoothing parameter for level, q is the smoothing parameter for trend, $\emptyset$ is the dampening parameter and $l_t = y_t$ for $t \leq p$, $b_t = \frac{y_t}{y_{t-1}}$ for $t \leq q$, $b_1 =$

$1, p \in \{1,2,\dots,n\}, q \in \{0,1,2,\dots,p\}, \emptyset \in (0,1]$. Then, the h step ahead forecasts can be obtained by:

$$\hat{y}_{t+h|t} = l_t + b_t^{(\emptyset+\emptyset^2+\cdots+\emptyset^h)}. \tag{3.6}$$

Since both versions of the method require three parameters we will distinguish between them by using the notation $ATA_{add}(p,q,\emptyset)$ for the additive form and $ATA_{mult}(p,q,\emptyset)$ for the multiplicative form.

Notice that when $q = 0$ both forms of ATA are reduced to the simple form $ATA(p,0,\emptyset)$ which can be written as:

$$\mathrm{l_t} = \left(\frac{\mathrm{p}}{\mathrm{t}}\right)\mathrm{y_t} + \left(\frac{\mathrm{t-p}}{\mathrm{t}}\right)\mathrm{l_{t-1}}, \tag{3.7}$$

where $p \in \{1,2,\dots,n\}$ and $l_t = y_t$ for $t \le p$. Forecasts then can be obtained by $\hat{y}_{t+h|t} = l_t$.

When $q \neq 0$ and $\emptyset = 1$ the additive and multiplicative forms of ATA are reduced to the trended versions $ATA_{add}(p,q,1)$ and $ATA_{mult}(p,q,1)$ which are given below respectively:

$$l_t = \left(\frac{p}{t}\right)y_t + \left(\frac{t-p}{t}\right)(l_{t-1} + b_{t-1}), \tag{3.8}$$

$$b_t = \left(\frac{q}{t}\right)(l_t - l_{t-1}) + \left(\frac{t-q}{t}\right)(b_{t-1}), \tag{3.9}$$

$$\hat{y}_{t+h|t} = l_t + hb_t\ , \tag{3.10}$$

and

$$l_t = \left(\frac{p}{t}\right)y_t + \left(\frac{t-p}{t}\right)(l_{t-1}b_{t-1}), \tag{3.11}$$

$$b_t = \left(\frac{q}{t}\right)\left(\frac{l_t}{l_{t-1}}\right) + \left(\frac{t-q}{t}\right)b_{t-1}, \tag{3.12}$$

$$\hat{y}_{t+h|t} = l_t + b_t^h. \tag{3.13}$$

Another distinction can be made based on the parameter optimization process used for these forms. Unless otherwise stated, the parameter values that minimized the in-

sample one step ahead sMAPE are used as optimum values and optimization is carried out for all the parameters simultaneously.

Obtaining the point forecasts:

1. The data sets are tested for stationarity using the Augmented Dickey-Fuller test using the function ndiffs from the forecast package in R.

2. The data sets are tested for seasonality using the SeasonalityTest function from the 4Thetamethod.R code downloaded from the competitions GitHub repository for $\alpha = 0.20$ and seasonality periods 4 for quarterly, 12 for monthly, 7 for daily. The data sets that could be classified as seasonal by this test are de-seasonalized by the classical multiplicative decomposition method.

3. Arbitrary models were used for different data types.

4. The forecasts are re-seasonalized using the seasonality indexes from the classical multiplicative decomposition to obtain final point forecasts. If any negative forecasts are obtained they are set equal to zero.

ATA(p,0) gives greater emphasis than SES to the most recent history and less emphasis than SES to the more distant past observations at the same smoothing constant. Adaptation of ATA(p,0) to any trended or level shifted data is faster than SES. Since ATA(p,0) gives greater emphasis to the most recent history versus the more distant past comparing to SES at the same alpha level.

The values of the unknown parameters and the initial values that minimize SSE

$$SSE = \sum_{t=1}^{n} \left(X_t - \hat{X}_t \right)^2 = \sum_{t=1}^{n} e_t^{\,2} . \qquad (3.14)$$

Both optimum smoothing parameter and initial value are obtained simultaneously. There is no ”initilization problem” for ATA(p,0) when either n or α small. ATA(p,0) does not violate the concept of exponential smoothing since the smallest weight

attached to the last observation is $\frac{1}{n}$ but SES may attain smoothing parameter less than $\frac{1}{n}$ and $\alpha_{ATA} = \frac{p}{n}$.

$$\frac{1}{n} \leq \alpha_{ATA} \leq 1, \tag{3.15}$$

$$. \; 0 \leq . \quad . \quad . \leq \frac{1}{n} \leq \alpha_{SES} \leq 1. \tag{3.16}$$

Considering all these features, it is seen that the ATA method was developed by taking into account the principles of the exponential smoothing method. However, unlike the ES method, it has many features that make the ATA method stand out in theory and practice. The main features that stand out are the simultaneous realization of the initial value and the best model optimization, the ability to give exponential, equal and linear weights to the observations, the ability to model all time series behaviors additively or multiplicatively with a single formula thanks to the parameter pool, its simple structure and high estimation performance.

CHAPTER FOUR
META LEARNING

Machine learning has revolutionized the field of artificial intelligence by enabling systems to learn from data and make predictions and decisions without being explicitly programmed. Traditional machine learning algorithms typically require large amounts of labeled data to satisfactorily perform a particular task. However, in real-world scenarios, the available data may be limited, making it impractical to train the model from scratch for each new task. This limitation has led to the emergence of a subfield called meta-learning, also known as "learning to learn".

Meta-learning aims to address the challenge of adapting and generalizing knowledge across multiple tasks or domains using experience from previous learning tasks. Meta-learning explores ways to acquire knowledge and self-optimize learning algorithms, rather than focusing solely on training models for specific tasks. The basic idea is to allow models to learn, making them more efficient, more adaptable, and able to generalize to new tasks with minimal data and training overhead.

The concept of meta-learning has its roots in various fields such as transfer learning, cognitive science, Bayesian inference, and optimization techniques (Wang, 2021). Transfer learning focuses on reusing knowledge learned from a source task to improve the performance of the target task. Cognitive science studies human learning processes and memory, providing insights into how people adapt and generalize knowledge in different situations. Bayesian inference provides a probabilistic framework for reasoning about uncertainty and making informed decisions in meta-learning scenarios. Optimization techniques play an important role in adapting model parameters and learning strategies to different tasks.

Applications of meta-learning span a wide variety of fields, including computer vision, natural language processing, robotics, healthcare, and recommender systems (Sivamayil, 2023). In the field of computer vision, meta-learning techniques have been used to improve image classification and object recognition in scenarios with limited labeled data. Natural language processing has benefited from meta-learning in tasks

such as text generation and sentiment analysis. In robotics, meta-learning facilitates rapid adaptation to new environments and tasks, enabling more flexible and autonomous systems. Medical applications range from personalized medicine to medical image analysis. Meta-learning helps make accurate predictions using limited patient-specific data. In addition, we use meta-learning techniques to improve our recommendation system and provide more accurate and personalized recommendations to our users.

Despite significant progress in meta-learning, some challenges remain (Vettoruzzo, 2023). Evaluating and benchmarking meta-learning algorithms is not a trivial task, as it requires the definition of suitable metrics and the creation of standardized evaluation protocols. Furthermore, ethical considerations such as bias and fairness must be considered when applying meta-learning to real-world applications. This thesis focuses on applying meta-learning approaches specifically to time series.

4.1 Meta Learning for Forecasting

Meta-learning for forecasting refers to a particular approach that focuses on gathering information gleaned from multiple related time series tasks to improve the performance of forecasting models. Models are trained on a large number of time series data sets in order to be able to adapt to, and make accurate predictions about, previously unknown new time series data (Prudêncio & Ludermir, 2004; Lemke & Gabrys, 2010; Talagala et al., 2018).

Meta-learning initially assumes that different time series data share some underlying patterns or characteristics. These patterns are seasonality, trends, or relationships between variables. Meta-learning identifies similarities between different time series tasks based on these common patterns.

In the next step, the metamodel is trained on a set of historical time series datasets, often called "base tasks". These base tasks can be different time series from different domains or different time series from the same domain. A metamodel learns how to extract common features or representations that help make predictions across different

tasks. Base tasks are taken from the M3 competition dataset. Metamodels may incorporate logistic regression models, decision trees, neural networks etc. To test the performance of metamodels, we need to quantify their performance using unseen data. As new time series tasks arise, the metamodel adapts the learned knowledge to that specific task. Based on the similarity found between the new task and the base task, we can adjust the internal parameters and adjust the prediction strategy.

To this end, we reserve part of the M3 competition series as new tasks and test meta-learners on this data. Finally, the metamodel can make predictions on new time series data. The idea is that the knowledge gained from the base task can help us to generalize better and generate more accurate predictions for new tasks, even when only limited data points are available. Chapter 5 compares meta-learner results with basic model selection procedures.

Meta-learning in time series forecasting is beneficial in scenarios where data for a particular time series is limited, or when you want to improve your time series forecasting model's generalization skills across different datasets. It basically uses the principle of transfer learning to make predictive models more adaptive and robust (Gastinger et al., 2021). To improve prediction accuracy, we use simple mean, then median, and finally logistic regression as meta-learning approaches.

4.2 Simple Averaging

Simple averaging of time series forecasts is a simple ensemble technique to improve the accuracy of time series forecasts and can be considered the simplest meta-learning model. The model combines forecasts from multiple forecasting methods by determining a simple average. This approach helps reduce the impact of individual model errors and improve overall predictive accuracy.

First, multiple forecasts are made from the same time series using different forecasting methods or models. These methods may include statistical models such as Ata, ARIMA, exponential smoothing, machine learning algorithms, or other suitable techniques. Each forecasting method provides its own forecast for future dates based

on historical data. These individual forecasts represent different perspectives on how the time series will evolve. To create an ensemble forecast, the individual forecasts are averaged for each future time point. Simply the predictions from all models for a given time step are aggregated and divided by the total number of predictions.

Mathematically, it can be represented as:

$$\hat{f}(t) = \frac{\sum_{i=1}^{N} \hat{f}_i(t)}{N}, \tag{4.1}$$

where $\hat{f}(t)$ is the ensemble forecast and $\hat{f}_i(t)$ for i=1,…,N are the individual forecasts obtained from single models.

The resulting ensemble prediction represents the average of the individual predictions and can be used as the final prediction. This combined forecast can often be more accurate and more robust than using a single forecasting method. Simple averaging works well when each model provides diverse and complementary insights into the time series data. Harnessing the collective wisdom of multiple models helps reduce the noise and errors associated with individual prediction methods.

Simple averaging in meta-learning has several advantages. First, it mitigates overfitting by combining knowledge gained from multiple models trained for different tasks. Due to the diversity of tasks, the ensemble reliably captures a wider range of patterns, leading to better generalization of unknown tasks. Furthermore, simple averaging reduces the impact of outliers and noisy predictions from individual models, increasing prediction robustness. It provides a form of regularization that helps mitigate biases and errors that may exist in individual models.

Simple averaging also has some limitations. We assume that the underlying model is equally competent and its predictions are equally reliable. In practice, some baseline models may perform better than others on a given task, and blind averaging of predictions may not provide optimal results. Techniques such as weighted averaging, which weight individual model contributions based on performance or reliability, can therefore be used to address this limitation.

In summary, simple averaging is a widely used meta-learning technique that uses ensemble predictions from multiple models trained for different tasks. Catching different patterns and reducing overfitting improves generalization and robustness. Simple averaging serves as a fundamental method of meta-learning and continues to be explored in various applications and extensions. Simple averaging is easy to implement and often effective, but it is not always the best ensemble technique for all time series forecasting problems.

Depending on the characteristics of the data and the prediction method used, more advanced ensemble techniques such as weighted averaging, stacking, or even machine learning-based approaches may yield even better results.

4.3 Using the Median of Forecasts

Combining forecasts by taking the median is also a very simple approach that works very well for many forecasting tasks. This idea has also been adopted by more complex approaches such as Simple Combining Univariate Models (SCUM) (Petropoulos & Svetunkov, 2020). Combined median forecasting is an ensemble forecasting technique used to improve forecast accuracy and robustness by obtaining the median forecast from multiple models or techniques. This is a variation of the simple mean method, but uses the median instead of the mean to combine the predictions.

The median approach is especially useful when dealing with time series data and other forecasting problems where outliers or extreme values can affect forecast accuracy. Like simple averages, time series forecasts are first created using different forecasting models or forecasting methods. These models are diverse and include statistical methods, machine learning algorithms, or other predictive approaches. Each forecasting method provides its own forecast for future points in time based on historical data.

To create an ensemble forecast, the median of the individual forecasts is calculated for each future time point. Median is the middle value in a sorted list of numbers,

meaning it is not affected by extreme outliers. Mathematically, for a given time step t, we can express

$$\hat{f}(t) = Median(\hat{f}_1(t), \hat{f}_2(t), \ldots, \hat{f}_N(t)) \,, \tag{4.2}$$

where $\hat{f}(t)$ is the ensemble forecast and $\hat{f}_i(t)$ for i=1,…,N are the individual predictions obtained from the individual models. The resulting ensemble prediction represents the median of the individual predictions and can be used as the final prediction.

This approach is particularly effective when the goal is to reduce the impact of outliers or extreme values that can skew results using a simple averaging approach. The median method is robust because it is less susceptible to outliers in each prediction. Even if one or more models produce predictions that differ significantly from the majority, these extreme values do not disproportionately influence the final ensemble prediction. This is especially important in situations where extreme events can lead to inaccurate forecasts if not properly managed. However, the median method cannot correct for the imprecision of the individual predictions themselves, so it is important to ensure that the combined individual predictions are based on reliable and well-validated models.

The two approaches discussed so far, the simple mean method and the median approach, work fairly well, but you can improve the prediction performance by applying a meta-learning model. A meta-learning model must identify the properties that define the similarities and relationships between individual data sets and how these properties affect the best performing method for each data set. We call these properties time series features. Some of the commonly used features are described in the next section.

4.4 Time Series Features

In the context of meta-learning, "features" refer to representations or characteristics of data that are learned or constructed to facilitate the meta-learning process. They play a key role in capturing task-specific information, allowing effective customization and

generalization across different learning tasks. They provide a compact and informative representation of the input data that can be used in meta-learning algorithms to extract and transfer task-independent knowledge to new tasks. A time series feature is a descriptive attribute or characteristic derived from time series data that provides valuable information about the underlying patterns, trends, and statistical characteristics of the data. Extracting meaningful features is an important step in time series meta-learning. Applications in this study utilize most of the techniques of Talagale et al. (2018). The authors proposed an approach called FFORMS (Feature-based Forecast Model Selection). It is a framework for automatic time series forecasting that uses a wide range of time series features to select the optimal forecasting model. These characteristics are important for model selection, and FFORMS uses these characteristics to determine which forecasting model performs best for a given set of time series data.

A problem considered in this way becomes a supervised classification problem. In this classification problem, we used the specific features from Table 2, following the approach of Talagale et al. (2018). The feature computations in Table 2 are performed in the R programming language, package tsfeatures.

Table 4.1 Specific Features

Name of Feature	Description
Trend	Strength of trend
Spike	Strength of spikiness
Linearity	Amount of linearity
Curvature	Amount of curvature
e_acf1	First ACF value of the remainder series
e_acf10	Sum of squares of first 10 ACF values of the remainder series

Table 4.1 Contiued Specific Features

Entropy	Spectral entropy
x_acf1	First ACF value of the original series
x_acf10	Sum of squares of first 10 ACF values of the original series
diff1_acf1	First ACF value of the differenced series
diff1_acf10	Sum of squares of first ten ACF values of differenced series
Type	Type of time series data (Macro, Micro, Demography, Finance, Industry, Other)
Stability	Amount of stability
Lumpiness	Amount of lumpiness
Hurst	The Hurst exponent
Nonlinearity	Amount of nonlinearity
Crossing points	The number of times a time series crosses the median
Flat spots	Maximum run length within any single interval out of ten equal sized intervals
Unitroot kpss	The statistic for the Kwiatkowski et al. (1992) unit root test
Unitroot pp	The statistic for the Phillips & Perron (1988) unit root test
Arch_acf	AR model gives new series y, the sum of squares of the first 12 autocorrelations of y^2

Table 4.1 Continued Specific Features

Garch_acf	AR model gives new series y, a GARCH(1,1) model is fitted to y and the residuals are e, the sum of squares of the first 12 autocorrelations of e^2
Arch_LM	A statistic based on the Lagrange Multiplier (LM) test of Engle (1982) for autoregressive conditional heteroscedasticity (ARCH) model
X_pacf5	The first PACF value of the original series
Diff1x_pacf5	The sum of squares of the first five PACF values of the differenced series
Diff2x_pacf5	The sum of squares of the first five PACF values of the twice differenced series

4.5 Feature Based Meta Learning Approaches for Time Series Forecasting

Feature-based meta-learning approaches to time series forecasting involve using features extracted from time series data to facilitate training and fitting forecasting models. These approaches aim to improve the performance of time series forecasting models by leveraging meta-learning principles to learn from different time series tasks and their associated features (Ma & Fildes, 2021).

First, a comprehensive set of features must be extracted from historical time series data. These features may include statistical, temporal, spectral, and other related attributes that capture features of the time series. The extracted features are transformed into suitable representations that can be used as input for machine learning models.

During the meta-training phase, a meta-learning model is trained using a collection of time-series tasks. Each task consists of a timeline and associated characteristics. The goal is to predict future values. Meta-learning models learn to map extracted features to prognostic model parameters or initial conditions. Our application does the mapping in such a way that the best performing time series forecasting model is selected. This mapping can be done using any suitable supervised learning algorithm.

During training, the model is subjected to different time-series tasks, enabling it to capture patterns, relationships, and similarities between tasks. In the metatest phase, the trained meta-learning model is evaluated using new, never-before-seen time-series tasks. Each of these tasks has its own characteristics and time series data. Our application also captures models with the best out-sampled performance.

The model uses the features extracted from the new task to tune the predictive model. This tuning may include fine-tuning model parameters and initializing model states. After fitting, the model makes predictions on new time series data. The forecasts of the fitted forecast model may be aggregated to produce the final forecast. Common aggregation methods include simple averages, weighted averages, or more advanced ensemble techniques. In our case, we will focus on choosing the best forecasting method for a particular time series. Therefore, the meta-learner is chosen as logistic regression. The model will be explained in the next section.

4.6 Binary Logistic Regression

Binary logistic regression (Cox, 1958) is a statistical model for binary classification. That is, it is designed to predict one of two possible outcomes, usually represented by 0 (failure) and 1 (success). Despite its name, logistic regression is a classification algorithm, not a regression algorithm. It is widely used in various fields such as biostatistics, finance and industry.

Logistic regression belongs to the family of generalized linear models (Hastie & Pregibon, 2017) and the key function in defining a logistic regression model is the sigmoid (σ) function, also known as the logistic function, which can be written as:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{4.3}$$

where z is a linear combination of the input features and model parameters: $z = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p$ where b represents the regression coefficients, x are the input features, and p is the number of features.

The sigmoid function maps all real numbers z to the range [0, 1]. This is important for binary classification since the output can be interpreted as probabilities thanks to this transformation. Logistic regression models the logarithm of the odds called the log-odds of the positive class (1) (also called the logit) which can be written as:

$$\log(Odds) = log\left(\frac{P(Y=1)}{P(Y=0)}\right) = z = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p. \tag{4.4}$$

So the logistic regression model basically tries to find the best linear relationship between the input features and the log odds of the positive class. Once this relationship has been estimated, researchers can proceed from linear combinations of features to probabilities of belonging to the positive class using the inverse function of the logit values (just the sigmoid function).

To train a logistic regression model, researchers need labeled data. This is because the model is a supervised algorithm with a binary (0 or 1) result and corresponding input characteristics. For time series applications, the input is the features extracted from the series, and the output equals 1 if a particular version of Ata is the best model, and 0 otherwise.

The goal during training is to find the optimal value of the coefficients that minimize the difference between the predicted labels using the predicted probability (σ(z)) and the actual label in the training data. A well-known estimation method called

maximum likelihood estimation is used for this purpose, but also iterative optimization with algorithms such as Newton-Raphson and Fisher's scoring are used (Agresti, 2012). These are typically done via statistical programming languages, and R is used in this work.

A positive estimate of the coefficient indicates that an increase in the corresponding trait leads to an increase in the log odds of the positive class. However, negative coefficients indicate that increasing traits decrease the log odds of belonging to the positive class. This relationship can be easily interpreted by taking the exponent of the estimate and using it as the odds ratio.

It is possible to predict Y using the estimated probabilities. To classify the data into one of the two classes, we need to determine the threshold. You can perform receiver operating curve (ROC) analysis to select the optimal threshold (Kleinbaum et al., 2010; Bilder & Loughin, 2014). The performance of the estimated logistic regression model can be evaluated using test data. A model's classification accuracy can be measured using various metrics such as sensitivity, specificity, accuracy, precision and F-score. For confusion matrices which summarize the models performance as in Table 2, these measures can be computed using the definitions in Table 3.

Table 4.2 An Example Confusion Matrix

		Predicted Label	
		Positive (1)	Negative (0)
Actual Label	Positive (1)	True Positive (TP)	False Negative (FN)
	Negative (0)	False Positive (FP)	True Negative (TN)

Table 4.3 Performane Measures

Measure of Performance	Formula
Accuracy	(TP+TN) / (TP+TN+FP+FN)
Sensitivity	(TP) / (TP+FN)
Specficity	(TN) / (TN+FP)
Precision	(TP) / (TP+FP)
F-score	2 (Sensitivity*Precision) / (Sensitivity+Precision

4.7 Classification Trees

Decision trees are well-known and widely used algortihms that aid in both classification and regression problems. Just like binary logistic regression, decision trees are used for determining the most suitable forecasting method for the time series data at hand in this thesis. For this reason, the focus will be on classification purposes. Decision trees are named as such since they literally look like upside down trees when visualized. The trees are formed with nodes and edges where nodes are basically tests on features and edges represent the outcomes of those tests (Quinlan, 1986).

Each decision tree starts with a root node which is a test performed on the most important feature (Quinlan, 1986). The inner nodes that follow this root node are called internal nodes. The final outcomes or predictions made by the tree are called the leaf nodes (Gordon et al., 1984). The sample space is therefore split into regions and for each region the most repeated class is calculated using training data and that class is used as the predicted class at terminal nodes.

To decide which nodes are needed and which features are more prominent, splitting criteria are used. Common criteria used for this purpose are information gain and Gini impurity (Quinlan, 1986). These measures are based on the concept of entropy which

is basicallt the measure of impurity in a dataset (Shannon, 1948) and the decisions on where to split and which features to use are made such that the best possible reduction on entropy is achieved. Entropy for a node of a classification tree can be defined as:

$$H(S) = -\sum_{i=1}^{k} p_i log_2(p_i), \tag{4.5}$$

where p_i is the proportion of data that are in class i and k is the total number of classes.

Whenever a split is considered, the weighted average entropy is calculated for that split. This is denoted by $H(S|A)$ which is simply the weighted average of the entropies of the subsets created by the feature. The weighting is done by using the proportion of data in that subset. Then, information gain (IG) can be calculated by:

$$IG(S, A) = H(S) - H(S|A) \tag{4.6}$$

The feature that results in greater IG is chosen in successive splits. Another measure used for this decision is the Gini index, which is used by the famous classification and regression tree (CART) algorithm. It can be written as:

$$Gini = 1 - \sum_{i=1}^{k} p_i^2 \, . \tag{4.7}$$

For some decision tree algorithms, the splits are made by choosing the feature with the smallest Gini index at each step. The resulting classificaiton tree may be different depending on the splitting criterion and tree building strategy (Tibshirani et al., 2017). Usually, a final tree is chosen so that the predictive performance is optimized. The decision trees are very popular since they are easy to interpret and easy to implement however overly grown trees may be very complex and overfit data. Pruning, stopping the tree at an internal step, is a strategy that is used quite often however a more powerful approach is using random forests which are ensembles of multiple decision trees (Breiman, 2001).

4.8 Random Forest

In order to overcome the overfitting problem of single classification trees and to improve predictive performance, random forests can be used for classification purposes. They are one of the most well known ensemble approaches which combine information from multiple models (Dietterich, 2000).

In a random forest, bootstrap samples (samples of same size chosen with replacement) are chosen from the training data and random subset of features are considered for splitting at each step when building the decision trees. These are the two main ideas behind random forests. Using bootstrap samples and not considering all features at each step. This introduces two hyper-parameters: how many trees will be built (N) and how many random features will be considered at each split (mtry) (Breiman, 2001). By building many trees that are meant to be different from each other (which is achieved by varying the candidate features at splits) the random forest uses the class that is chosen the most by the N trees as its prediction. The random forest uses the data points that are not included in each bootstrap sample (the out-of-bag data dented as OOB) as validation sets and the misclassification rate is calculated for these data and averaged to get the OOB error rate. The random forest can also provide a measure of the contribution of each feature to the final decision called the feature importance.

By fine-tuning the hyper-parameters, the random forest models provide much better fit compared to single trees and they the risk of overfitting problems. The fact that they provide OOB estimates of error and feature importance also maket hem desirable mashine learning approaches (Tibshirani, 2017).

In our applications, the most suitable forecating model for a time series is predicted using a random forest model and the results are compared to those from simpler approaches and logistic regression.

CHAPTER FIVE
APPLICATION

The time series data from the M3-Competition is used for demonstrating the methodology presented in this thesis. The data consists of 3003 separate time series where each series consist of in and out sample sections. The results from simple averaging and using the median of forecasts are summarized in the following sections. For comparing the forecasting accuracy of different approaches, first simpler meta-learning approaches are applied.

Next, feature based meta-learning approaches are applied. Here, the M3 series are analyzed separately depending on their frequencies, i.e the results are obtained separately for yearly, quarterly, monthly and other data. The results obtained by running logistic regression and random forest analysis are presented in the last section.

5.1 Results from Simple Averaging and Median Forecasts

The forecasting performance of any method or model may vary according to the statistical techniques used. Before ATA method is applied, the data sets were deseasonalized by the classical multiplicative decomposition method, when necessary. The parameters are optimized by minimizing the in-sample one-step-ahead acoording to selected accuracy measure (sMAPE, MSE, MAE, MAPE, MASE) for all forecast horizons are computed and averaged across all 3003 series. Results from three different applications of the ATA method will be considered here.

(i) ATA(p, 0) where p is optimized for q = 0

(ii) ATA(p, 1) where p is optimized for q = 1

(iii) ATA − comb where a simple average of the forecasts from the two models ATA (p, 0) and ATA(p, 1).

The 3003 series of the M3-Competition include various types of time series data (micro, industry, macro, etc.) and different time intervals between successive observations (yearly, quarterly, monthly, etc.). The rankings of the performance of the various methods vary according to the accuracy measure being used. For this reason, the forecasts were obtained by using different accuracy measures and their out-sample performances are measured according to symmetric mean absolute error. According to the results given in Table 5.1, performances may vary according to the time intervals and accuracy measures. The accuracy of the combination of various accuracy measures outperforms, on average, the specific methods being combined and does well in comparison with other methods. As can be seen in the Table 5.1, the best overall result is obtained with the simple averağe of forecast. Ata (p, 0), which is an alternative to the SES, gives better results no matter which accuracy measure we use.

Table 5.1 Average symmetric MAPE of ATA(p,0) for different accuracy measures

	Yearly	Quarterly	Monthly	Other	Overall
sMAPE	18.00	9.71	14.49	6.26	13.77
MAE	17.90	9.66	14.49	6.26	13.75
MSE	17.87	9.78	14.65	6.26	13.88
MAPE	18.07	9.88	14.61	6.36	13.89
MASE	17.90	9.89	14.21	6.26	13.59
Simple Average of Forecasts	**17.80**	9.78	**14.24**	6.29	**13.58**
Median of Forecasts	17.91	**9.70**	14.42	**6.26**	13.70
SES	17.82	9.72	15.32	6.29	14.32

When we apply the trended form of the Ata method with *q=1*, it again gives better results than its counter form Holt exponential smoothing model, in the overall. However, best forecasting results are obtained when sMAPE is used alone as it seen in Table 5.2.

However, The best forecast results of the Ata method given in Table 5.3 are obtained by averaging the forecasts of the two models *Ata (p, 0)* and *ATA(p, 1)* according to different accuracy measures. The results obtained here for ATA method are better than the Theta method winner of the 3003 M-competition and its counter exponential smoothing models.

Table 5.2 Average symmetric MAPE of Ata(p, 1) for different accuracy measures

	Yearly	Quarterly	Monthly	Other	Overall
sMAPE	16.78	9.24	15.33	4.87	14.09
MAE	16.73	9.19	15.72	4.88	14.35
MSE	16.76	9.31	16.14	4.89	14.67
MAPE	16.91	9.47	16.24	4.94	14.78
MASE	16.73	9.48	15.75	4.88	14.42
Simple Average of Forecasts	16.80	9.39	15.56	5.00	**14.28**
Median of Forecasts	**16.65**	**9.22**	15.67	4.88	14.31
Holt	19.27	10.67	**15.36**	**4.81**	14.60

Table 5.3 Average symmetric MAPE of combination Ata(p, 0) and Ata(p, 1) for different accuracy measures

	Yearly	Quarterly	Monthly	Other	Overall
sMAPE	16,54	9,07	13,75	4,94	12,94
MAE	16,49	9,04	13,78	4,95	12,96
MSE	16,48	9,15	13,93	4,95	13,08
MAPE	16,63	9,27	13,90	5,03	13,10
MASE	16,49	9,29	13,67	4,95	12,92
Simple Average of Forecasts	**16,42**	9,18	**13,56**	5,00	**12,82**
Median of Forecasts	16,50	9,10	13,62	5,01	12,86
Holt	19,27	10,67	15,36	4,81	14,60
S-H-D	17,07	9,22	14,48	4,56	13,52
Theta	16,90	**8,96**	13,85	**4,41**	13,01

5.2 Results from Feature Based Meta-Learning Approaches

For data preparation, the M3-competition data needed to be transformed into a supervised learning data. Different forecasting approaches need to be used as base-learners, features need to be extracted and finally meta-learning model needs to be fitted. For this purpose, two versions of Ata method, namely the $Ata(p, q = 0, \emptyset)$ which is the simple form and $Ata_{Add}\,(p, q = 1, \emptyset = 1)$ which is the equally weighted additive trended form are chosen as base learners. The former model is denoted by 0

and the latter model is denoted by 1. The output variable for our supervised learning problem is then the model which has the smallest out-sample forecast accuracy where this variable will take on the value of 0 if for that particular data simple form works better and will be equal to 1 if the trended version provides better forecasts.

The input for the supervised learning problem consists of the time series features defined in Table 4.1 in Chapter 4. For each of the 3003 time-series these features are calculated and saved in addition the average in sample sMAPE performances of the two models.

After these preparations, 3003 time-series are first categorized according to frequency and for each type of data training and test data are obtained by randomly splitting the data (80%-20%). Initially logistic regression model is used as the meta-learner for explaining the relationship between the optimal model and features. For fitting the logistic regression model, stepwise variable selection is applied. The cut off for the probabilities for class prediction are determined using ROC analysis. The cut off values are 0.5 for yearly, 0.6 for quarterly, 0.5 for monthly and 0.5 for other. Next, random forest is used as the meta learner. For the random forest applications, the number of variables to be considered at each split is left as default and number of trees is set to 1000. The results from these approaches are compared in terms of the proportion of correctly identifying the optimal model on the test data. The comparisons are also conducted comparing the findings from these to models versus naively choosing the optimal model as the model with smaller in-sample sMAPE for complete data.

All the computations are carried out in R programming and the packages named tsfeatures, Mcomp, ATAforecasting, tidyverse, caret, MASS and randomForest are used. Example code is provided in the Appendix.

Table 5.4 Accuracies of learners on identifying optimal forecaster

Data frequency	Naïve selection	Logistic Learner	Random Forest Learner
Yearly	0.6248	0.7132	0.7132
Quarterly	0.5397	0.6645	0.7039
Monthly	0.5707	0.6678	0.7063
Other	0.7529	0.7429	0.7714

The accuracy values in Table 5.4 are calculated as follows. For the naïve selection method, the in-sample sMAPE values of the two Ata models are compared and the one with the smallest error is chosen. This decision is compared to the actual optimal selection determined by comparing the out-sample sMAPE values of the two models. The accuracy value for naïve selection gives the proportion of correct decisions by the algorithm for all yearly, quarterly, monthly and other series.

For the logistic learner column, the binary logistic regression models are built using all the time-series features in addition to in-sample sMAPE values of the two Ata models. Model selection is carried out using stepwise algorithm and only training data is used for coefficient estimation. The trained model is tested on reserved test data where the predictions from the model are compared actual optimal forecasters. The accuracies are the proportions of the logistic learner choosing the right model on test data.

The random forest approach used in this thesis in an ensemble of 1000 trees and 5 variables are considered at each split. Like logistic regression, the features considered for the random forest are the previously defined time series features and in-sample sMAPE values of the two Ata models. The optimal model predictions from the random forest model are compared to the actual best performing models for test data and the accuracy column represents the propotion of correct decisions obtained in this manner.

As it can be seen from Table 5.4 for yearly data the feature based meta-learning approaches are equivalent and both perform better than the naïve selection. For quarterly and monthly data naïve selection works very poorly in identifying the best model and both logistic learner and random forest enhance the decision making process. Specifically, the random forest model provides much better results for both the quarterly and monthly time series data. For the other data, all of the approaches provide desirable model selection however the random forest mode still provides slightly better results

CHAPTER SIX
RESULTS AND CONCLUSIONS

Time series forecasting is a fundamental step for many branches of science. There have been many advancements in this field since the initial proposals of simple methods. With the introduction of machine learning, forecasting also benefited from the new points of view like other fields of statistics.

Classical approaches consisted of choosing the best possible forecasting method for the data set at hand depending on just the given data. The approaches included looking at in sample error measures and proceeding with the forecasting method that produced the minimum error rate. Later on, the idea of combining forecasts was proposed and researchers benefited from taking the average, median or some weighted average of forecasts from different models.

With the help of machine learning, feature based meta-learning approaches were developed. These methods provide many advantages. Feature-based meta-learning enables the transfer of knowledge and patterns learned from one-time series task to another, even when the tasks come from different domains or have different characteristics. By adapting forecasting models based on task-specific features, meta-learning models can potentially make more accurate predictions for diverse time series data. Meta-learning can enhance the robustness of forecasting models, making them less susceptible to overfitting or poor performance on specific datasets. Meta-learning can reduce the need for extensive hyperparameter tuning for individual time series tasks since it learns to adapt automatically.

Overall, feature-based meta-learning approaches offer a promising way to enhance the performance and generalization capabilities of time series forecasting models, particularly when dealing with a wide range of time series datasets with varying characteristics. This was proven by the applications in this thesis where the accuracy (i.e., the proportion of actually choosing the best forecasting method for data sets) was increased by using feature-based meta learning methods.

In our applications, logistic regression model was used as a meta-learner. Logistic regression is a simple yet powerful algorithm for binary classification. Its simplicity

makes it easy to interpret and implement. However, it assumes that the relationship between the features and the log-odds is linear, which may not always be the case in real-world problems. In such cases, more complex models like decision trees, support vector machines, or neural networks may be considered. Out of these models we considered random forests. Our applications showed that the random forest learner was slightly better than logistic regression in identifying the best forecasting method and both meta-learners provided significant improvements compared to the naïve selection.

REFERENCES

Assimakopoulos, V., & Nikolopoulos, K. (2000). The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, *16*(4), 521–530.

Brown, R. G. (1959). *Statistical forecasting for inventory control.* New York: McGraw/Hill.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Cetin, B., & Yavuz, I. (2020). Comparison of forecast accuracy of ata and exponential smoothing. *Journal of Applied Statistics*, 1–11.

De Gooijer, J. G., & Hyndman, R. J. (2005). 25 years of iif time series forecasting: A selective review. *Tinbergen Institute Discussion Papers No. TI*, 05–068.

Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting, 4*(1), 1–28.

Gardner, E. S. (2006). Exponential smoothing: The state of the art—part ii. *International Journal of Forecasting*, *22*(4), 637–666.

Gardner Jr, E. S., & McKenzie, E. (1985). Forecasting trends in time series. *Management Science, 31*(10), 1237–1246.

Goodwin, P. (2010). The holt-winters approach to exponential smoothing: 50 years old and going strong. Foresight: The International Journal of Applied Forecasting,30–33.

Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Biometrics, 40(3), 874.

Holt, C. (1957). Forecasting trends and seasonal by exponentially weighted averages. *International Journal of Forecasting, 20*(1), 5–10. 46

Hyndman, R. J., & Athanasopoulos, G. (2014). *Forecasting: principles and practice*. Australia: OTexts.

Hyndman, R. J., Billah B. 2003. Unmasking the Theta method. *International Journal of Forecasting 19*(2), 287-290.

Hyndman, R. J., et al. (2006). Another look at forecast-accuracy metrics for intermittent demand. Foresight: *The International Journal of Applied Forecasting,4*(4), 43–46.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*(4), 679–688.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting, 18*(3), 439–454.

Koning, A. J., Franses, P. H., Hibon, M., & Stekler, H. O. (2005). The m3 competition: Statistical tests of the results. *International Journal of Forecasting, 21*(3), 397–409.

Makridakis, S., & Hibon, M. (2000). The m3-competition: results, conclusions and implications. *International Journal of Forecasting, 16*(4), 451–476.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting, 34*(4), 802–808. 47

Makridakis, S. G., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., & Winkler, R. (1984). *The forecasting accuracy of major time series methods.* New York: Wiley.

Pegels, C. C. (1969). On startup or learning curves: An expanded view. *AIIE Transactions, 1*(3), 216–222.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning, 1*, 81-106.

Taylan, A. S., Yapar, G., & Selamlar, H. T. (2021). Automatic Time Series Forecasting with Ata Method in R: ATAforecasting Package. R Journal, 13(2).

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379-423.

Taylor, J. W. (2003). Exponential smoothing with a damped multiplicative trend. *International Journal of Forecasting,* 19(4), 715–725.

Tibshirani, H. R., James, G., & Trevor, D. W. (2017). *An introduction to statistical learning*. Springer publication.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3), 324–342.

Yapar, G., Capar, S., Selamlar, H. T., & Yavuz, I. (2018). Modified holt's linear trend method. *Hacettepe Journal of Mathematics and Statistics, 47*(5), 1394–1403.

Yapar, G., Selamlar, H. T., Capar, S., & Yavuz, I. (2019). Ata method. *Hacettepe Journal of Mathematics and Statistics, 48*(6), 1838–1844.

Yapar, G., Yavuz, I., & Selamlar, H. T. (2017). Why and how does exponential smoothing fail? an in depth comparison of ata-simple and simple exponential smoothing. *Turkish Journal of Forecasting*, *1*(1), 30–39.